

VINNY TROIA

HUNTING CYBER CRIMINALS

A Hacker's Guide to
Online Intelligence Gathering
Tools and Techniques



 **SYBEX**
A Wiley Brand

Hunting Cyber Criminals



Hunting Cyber Criminals

A Hacker's Guide to
Online Intelligence Gathering
Tools and Techniques

Vinny Troia, PhD

WILEY

Copyright © 2020 by John Wiley & Sons, Inc., Indianapolis, Indiana
Published simultaneously in Canada

ISBN: 978-1-119-54092-2

ISBN: 978-1-119-54089-2 (ebk)

ISBN: 978-1-119-54099-1 (ebk)

Manufactured in the United States of America

No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise, except as permitted under Sections 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01923, (978) 750-8400, fax (978) 646-8600. Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, (201) 748-6011, fax (201) 748-6008, or online at <http://www.wiley.com/go/permissions>.

Limit of Liability/Disclaimer of Warranty: The publisher and the author make no representations or warranties with respect to the accuracy or completeness of the contents of this work and specifically disclaim all warranties, including without limitation warranties of fitness for a particular purpose. No warranty may be created or extended by sales or promotional materials. The advice and strategies contained herein may not be suitable for every situation. This work is sold with the understanding that the publisher is not engaged in rendering legal, accounting, or other professional services. If professional assistance is required, the services of a competent professional person should be sought. Neither the publisher nor the author shall be liable for damages arising herefrom. The fact that an organization or Web site is referred to in this work as a citation and/or a potential source of further information does not mean that the author or the publisher endorses the information the organization or website may provide or recommendations it may make. Further, readers should be aware that Internet websites listed in this work may have changed or disappeared between when this work was written and when it is read.

For general information on our other products and services please contact our Customer Care Department within the United States at (877) 762-2974, outside the United States at (317) 572-3993 or fax (317) 572-4002.

Wiley publishes in a variety of print and electronic formats and by print-on-demand. Some material included with standard print versions of this book may not be included in e-books or in print-on-demand. If this book refers to media such as a CD or DVD that is not included in the version you purchased, you may download this material at <http://booksupport.wiley.com>. For more information about Wiley products, visit www.wiley.com.

Library of Congress Control Number: 2019940773

Trademarks: Wiley and the Wiley logo are trademarks or registered trademarks of John Wiley & Sons, Inc. and/or its affiliates, in the United States and other countries, and may not be used without written permission. All other trademarks are the property of their respective owners. John Wiley & Sons, Inc. is not associated with any product or vendor mentioned in this book.

To my beautiful daughter, Aria, and my incredible wife, Jess. I never realized what a joy it would be to become a father, and I am thankful every day that you have both given me the most amazing gift of my life.



About the Author

Vinny Troia, PhD, CEH, CHFI, currently serves as head of Night Lion Security, a St. Louis–based cybersecurity consulting firm dedicated to providing top-tier ethical hacking and risk management services.

Troia has been recognized as a thought leader in cybersecurity and has become a go-to media expert for security-related discussions about major corporate data breaches, cyber law and legislation, airline and automobile hacking, and cyber-related scandals.

His experience in IT security stems from a lifetime of coding, complex problem solving, and self-taught computer skills. Troia now travels the globe speaking at conferences and security-related events and spends most of his free time hunting for data breaches and infiltrating private criminal circles on the darkweb.

With each new breach, valuable clues are left behind as to the evolution of an attacker’s methods. During his speeches, Troia uses that information to teach and inform others on ways to increase their defenses and put necessary response strategies in place for when incidents do occur.

Prior to starting Night Lion Security, Troia spent nearly a decade working on security- and risk-related projects for the U.S. Department of Defense.

Troia holds a PhD from Capella University and is a Certified Ethical Hacker and Certified Hacking Forensic Investigator.

For more information, including samples of Troia’s talks, please visit www.vinnytroia.com.

You can also connect with Vinny on LinkedIn at <https://linkedin.com/in/vinnytroia> or via Twitter at <http://www.twitter.com/vinnytroia>.



About the Technical Editor

Rhia Dancel conducts information security assessments throughout the United States, focusing on OSINT and risk-based management platforms with key engagements within the DoD and private sector space.

Rhia's technical and analytical background originated from a chemistry degree applied within the pharmaceutical industry for over 15 years. Rhia now supports organizations in their effort to implement security controls and achieve information security objectives across multiple security programs. Rhia also continues to provide technical input on risk- and security-based projects.



Acknowledgments

I would like to acknowledge and graciously thank the following people:

My Wife, for putting up with my countless sleepless nights and non-stop obsessing while I worked to crack this puzzle.

Bev Robb, without you, I don't think I would have been able to solve the mystery of TDO. Sometimes the most random connections and pieces of information can lead to the most significant discoveries, and that is exactly what happened. Thank you so much for putting up with my millions of questions and late-night text messages. I am eternally grateful to you and hope I can one day repay the favor.

Christopher Meunier, for never letting go of that easily identifiable chip on your shoulder that relentlessly muttered statements like this, giving me all the motivation I needed to keep pressing on:

*whitepacket@xmpp.is: I'm sorry man but you sound like you're either LE or a f***ing retard, probably the latter.*

Dennis Karvouniaris, Thanks for all the info and help you gave me along the way. I'm sorry things had to work out this way. I always enjoy our chats and hope that by the time you read this you will have taken my advice.

Chris "The Human Hacker" Hadnagy, for believing in me enough to connect me with the fine folks at Wiley, ultimately landing me this book deal.

Alex Heid and Jesse Burke, for all the back and forth and continued help pulling some of these pieces together.

And to all of this book's guest experts, I can't thank you enough for volunteering your time to contribute your stories and opinions for this book. I will be giving you all proper credits in the first chapter, but I had to give you all an extra shout-out here as well.



Contents at a Glance

Prologue		xxv
Chapter 1	Getting Started	1
Chapter 2	Investigations and Threat Actors	19
Part I	Network Exploration	43
Chapter 3	Manual Network Exploration	45
Chapter 4	Looking for Network Activity (Advanced NMAP Techniques)	67
Chapter 5	Automated Tools for Network Discovery	83
Part II	Web Exploration	119
Chapter 6	Website Information Gathering	121
Chapter 7	Directory Hunting	143
Chapter 8	Search Engine Dorks	159
Chapter 9	WHOIS	175
Chapter 10	Certificate Transparency and Internet Archives	201
Chapter 11	Iris by DomainTools	221
Part III	Digging for Gold	243
Chapter 12	Document Metadata	245
Chapter 13	Interesting Places to Look	267
Chapter 14	Publicly Accessible Data Storage	293

Part IV	People Hunting	323
Chapter 15	Researching People, Images, and Locations	325
Chapter 16	Searching Social Media	349
Chapter 17	Profile Tracking and Password Reset Clues	377
Chapter 18	Passwords, Dumps, and Data Viper	407
Chapter 19	Interacting with Threat Actors	433
Chapter 20	Cutting through the Disinformation of a 10-Million-Dollar Hack	453
	Epilogue	483
	Index	487



Contents

Prologue	xxv
Chapter 1 Getting Started	1
Why This Book Is Different	2
What You Will and Won't Find in This Book	2
Getting to Know Your Fellow Experts	3
A Note on Cryptocurrencies	4
What You Need to Know	4
Paid Tools and Historical Data	5
What about Maltego?	5
Prerequisites	5
Know How to Use and Configure Linux	5
Get Your API Keys in Order	6
Important Resources	6
OSINT Framework	6
OSINT.link	6
IntelTechniques	7
Termbin	8
Hunchly	9
Wordlists and Generators	9
SecLists	9
Cewl	10
Crunch	10
Proxies	10
Storm Proxies (Auto-Rotating)	10
Cryptocurrencies 101	11
How Do Cryptocurrencies Work?	12
Blockchain Explorers	13

	Following the Money	15
	Identifying Exchanges and Traders	17
	Summary	18
Chapter 2	Investigations and Threat Actors	19
	The Path of an Investigator	19
	Go Big or Go Home	20
	The Breach That Never Happened	21
	What Would You Do?	22
	Moral Gray Areas	24
	Different Investigative Paths	25
	Investigating Cyber Criminals	26
	The Beginning of the Hunt (for TDO)	27
	The Dark Overlord	27
	List of Victims	28
	A Brief Overview	29
	Communication Style	30
	Group Structure and Members	30
	Cyper	31
	Arnie	32
	Cr00k (Ping)	35
	NSA (Peace of Mind)	36
	The Dark Overlord	38
	Summary	41
Part I	Network Exploration	43
Chapter 3	Manual Network Exploration	45
	Chapter Targets: Pepsi.com and Cyper.org	46
	Asset Discovery	46
	ARIN Search	47
	Search Engine Dorks	48
	DNSDumpster	49
	Hacker Target	52
	Shodan	53
	Censys (Subdomain Finder)	56
	Censys Subdomain Finder	56
	Fierce	57
	Sublist3r	58
	Enumall	59
	Results	60
	Phishing Domains and Typosquatting	61
	Summary	64
Chapter 4	Looking for Network Activity (Advanced NMAP Techniques)	67
	Getting Started	67
	Preparing a List of Active Hosts	68
	Full Port Scans Using Different Scan Types	68
	TCP Window Scan	70
	Working against Firewalls and IDS	70

	Using Reason Response	71
	Identifying Live Servers	71
	Firewall Evasion	73
	Distributed Scanning with Proxies and TOR	73
	Fragmented Packets/MTU	74
	Service Detection Trick	74
	Low and Slow	76
	Bad Checksums, Decoy, and Random Data	76
	Firewalking	79
	Comparing Results	79
	Styling NMAP Reports	81
	Summary	82
Chapter 5	Automated Tools for Network Discovery	83
	SpiderFoot	84
	SpiderFoot HX (Premium)	91
	Intrigue.io	95
	Entities Tab	96
	Analyzing uberpeople.net	99
	Analyzing the Results	104
	Exporting Your Results	105
	Recon-NG	107
	Searching for Modules	111
	Using Modules	111
	Looking for Ports with Shodan	115
	Summary	116
Part II Web Exploration		119
Chapter 6	Website Information Gathering	121
	BuiltWith	121
	Finding Common Sites Using Google Analytics Tracker	123
	IP History and Related Sites	124
	Webapp Information Gatherer (WIG)	124
	CMSMap	129
	Running a Single Site Scan	130
	Scanning Multiple Sites in Batch Mode	130
	Detecting Vulnerabilities	131
	WPScan	132
	Dealing with WAFs/WordPress Not Detected	136
	Summary	141
Chapter 7	Directory Hunting	143
	Dirhunt	143
	Wfuzz	146
	Photon	149
	Crawling a Website	151
	Intrigue.io	152
	Summary	157

Chapter 8	Search Engine Dorks	159
	Essential Search Dorks	160
	The Minus Sign	160
	Using Quotes	160
	The site: Operator	161
	The intitle: Operator	161
	The allintitle: Operator	162
	The filetype: Operator	162
	The inurl: Operator	163
	The cache: Operator	165
	The allinurl: Operator	165
	The filename: Operator	165
	The intext: Operator	165
	The Power of the Dork	166
	Don't Forget about Bing and Yahoo!	169
	Automated Dorking Tools	169
	Inurlbr	169
	Using Inurlbr	171
	Summary	173
Chapter 9	WHOIS	175
	WHOIS	175
	Uses for WHOIS Data	176
	Historical WHOIS	177
	Searching for Similar Domains	177
	Namedroppers.com	177
	Searching for Multiple Keywords	179
	Advanced Searches	181
	Looking for Threat Actors	182
	Whoisology	183
	Advanced Domain Searching	187
	Worth the Money? Absolutely	188
	DomainTools	188
	Domain Search	188
	Bulk WHOIS	189
	Reverse IP Lookup	189
	WHOIS Records on Steroids	190
	WHOIS History	192
	The Power of Screenshots	193
	Digging into WHOIS History	193
	Looking for Changes in Ownership	194
	Reverse WHOIS	196
	Cross-Checking <i>All</i> Information	197
	Summary	199
Chapter 10	Certificate Transparency and Internet Archives	201
	Certificate Transparency	201
	What Does Any of This Have to Do with Digital Investigations?	202
	Scouting with CTFR	202

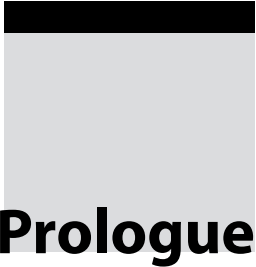
	Crt.sh	204
	CT in Action: Side-stepping Cloudflare	204
	Testing More Targets	208
	CloudFlair (Script) and Censys	209
	How Does It Work?	210
	Wayback Machine and Search Engine Archives	211
	Search Engine Caches	212
	CachedView.com	214
	Wayback Machine Scraper	214
	Enum Wayback	215
	Scraping Wayback with Photon	216
	Archive.org Site Search URLs	217
	Wayback Site Digest: A List of Every Site URL Cached by Wayback	219
	Summary	220
Chapter 11	Iris by DomainTools	221
	The Basics of Iris	221
	Guided Pivots	223
	Configuring Your Settings	223
	Historical Search Setting	224
	Pivootttt!!!	225
	Pivoting on SSL Certificate Hashes	227
	Keeping Notes	228
	WHOIS History	230
	Screenshot History	232
	Hosting History	232
	Bringing It All Together	234
	A Major Find	240
	Summary	241
Part III	Digging for Gold	243
Chapter 12	Document Metadata	245
	Exiftool	246
	Metagoofil	248
	Recon-NG Metadata Modules	250
	Metacrawler	250
	Interesting_Files Module	252
	Pushpin Geolocation Modules	254
	Intrigue.io	257
	FOCA	261
	Starting a Project	262
	Extracting Metadata	263
	Summary	266
Chapter 13	Interesting Places to Look	267
	TheHarvester	268
	Running a Scan	269
	Paste Sites	273

Psbtmp.ws	273
Forums	274
Investigating Forum History (and TDO)	275
Following Breadcrumbs	276
Tracing Cyper's Identity	278
Code Repositories	280
SearchCode.com	281
Searching for Code	282
False Negatives	283
Gitrob	284
Git Commit Logs	287
Wiki Sites	288
Wikipedia	289
Summary	292
Chapter 14 Publicly Accessible Data Storage	293
The Exactis Leak and Shodan	294
Data Attribution	295
Shodan's Command-Line Options	296
Querying Historical Data	296
CloudStorageFinder	298
Amazon S3	299
Digital Ocean Spaces	300
NoSQL Databases	301
MongoDB	302
Robot 3T	302
Mongo Command-Line Tools	305
Elasticsearch	308
Querying Elasticsearch	308
Dumping Elasticsearch Data	311
NoScape	311
MongoDB	313
Elasticsearch	314
Scan	314
Search	315
Dump	317
MatchDump	317
Cassandra	318
Amazon S3	320
Using Your Own S3 Credentials	320
Summary	321
Part IV People Hunting	323
Chapter 15 Researching People, Images, and Locations	325
PIPL	326
Searching for People	327
Public Records and Background Checks	330

Ancestry.com	331
Threat Actors Have Dads, Too	332
Criminal Record Searches	332
Image Searching	333
Google Images	334
Searching for Gold	335
Following the Trail	335
TinEye	336
EagleEye	340
Searching for Images	340
Cree.py and Geolocation	343
Getting Started	343
IP Address Tracking	346
Summary	347
Chapter 16 Searching Social Media	349
OSINT.rest	350
Another Test Subject	355
Twitter	357
SocialLinks: For Maltego Users	358
Skiptracer	361
Running a Search	361
Searching for an Email Address	361
Searching for a Phone Number	364
Searching Usernames	366
One More Username Search	368
Userrecon	370
Reddit Investigator	372
A Critical “Peace” of the TDO Investigation	374
Summary	375
Chapter 17 Profile Tracking and Password Reset Clues	377
Where to Start (with TDO)?	377
Building a Profile Matrix	378
Starting a Search with Forums	379
Ban Lists	381
Social Engineering	381
SE’ing Threat Actors: The “Argon” Story	383
Everyone Gets SE’d—a Lesson Learned	387
The End of TDO and the KickAss Forum	388
Using Password Reset Clues	390
Starting Your Verification Sheet	391
Gmail	391
Facebook	393
PayPal	394
Twitter	397
Microsoft	399

Instagram	400
Using jQuery Website Responses	400
ICQ	403
Summary	405
Chapter 18 Passwords, Dumps, and Data Viper	407
Using Passwords	408
Completing F3ttywap's Profile Matrix	409
An Important Wrong Turn	412
Acquiring Your Data	413
Data Quality and Collections 1–5	413
Always Manually Verify the Data	415
Where to Find Quality Data	420
Data Viper	420
Forums: The Missing Link	421
Identifying the Real "Cr00k"	422
Tracking Cr00k's Forum Movements	423
Timeline Analysis	423
The Eureka Moment	427
Vanity over OPSEC, Every Time	429
Why This Connection Is Significant	429
Starting Small: Data Viper 1.0	430
Summary	431
Chapter 19 Interacting with Threat Actors	433
Drawing Them Out of the Shadows	433
Who Is WhitePacket?	434
The Bev Robb Connection	435
Stradinatras	436
Obfuscation and TDO	437
Who Is Bill?	439
So Who Exactly Is Bill?	440
YoungBugsThug	440
How Did I Know It Was Chris?	441
A Connection to Mirai Botnet?	442
Why Was This Discovery So Earth-Shattering?	444
Question Everything!	445
Establishing a Flow of Information	446
Leveraging Hacker Drama	447
Was Any of That Real?	448
Looking for Other Clues	449
Bringing It Back to TDO	450
Resolving One Final Question	451
Withdrawing Bitcoin	451
Summary	452

Chapter 20	Cutting through the Disinformation of a 10-Million-Dollar Hack	453
	GnosticPlayers	454
	Sites Hacked by GnosticPlayers	456
	Gnostic's Hacking Techniques	457
	GnosticPlayers' Posts	459
	GnosticPlayers2 Emerges	461
	A Mysterious Third Member	462
	NSFW/Photon	463
	The Gloves Come Off	464
	Making Contact	465
	Gabriel/Bildstein aka Kuroi'sh	465
	Contacting His Friends	467
	Weeding through Disinformation	468
	Verifying with Wayback	468
	Bringing It All Together	469
	Data Viper	469
	Trust but Verify	472
	Domain Tools' Iris	474
	Verifying with a Second Data Source	475
	The End of the Line	476
	What Really Happened?	476
	Outofreach	476
	Kuroi'sh Magically Appears	477
	What I Learned from Watching Lost	477
	Who Hacked GateHub?	478
	Unraveling the Lie	479
	Was Gabriel Involved? My Theory	479
	Gabriel is Nclay: An Alternate Theory	479
	All roads lead back to NSFW	480
	Summary	481
Epilogue		483
Index		487



Prologue

One of the more recent investigations I worked on involved the hack of a multi-billion dollar organization. Their stolen data was posted for sale in private circles, and upon finding this out, I immediately contacted the organization. The organization had many questions, and given my prior investigative work, I was able to reach out to the threat actor on their behalf and obtain information on how the breach occurred.

The following text is a portion of the writeup provided by NSFW, *a threat actor we will be covering in much greater detail throughout this book*, where he describes, in detail, how he was able to hack this organization's network. The process he used was sophisticated, and by no means a run-of-the-mill drive-by hack.

This was very well planned and executed.

All identifying information has been changed.

HACK WRITEUP: NSFW

Firstly I realised that GitHub is adding new device verification within the week, therefore I tried to identify as many developers as possible and sign into their GitHub account to access organisation private repo's.

I then identified software developers working for Company using LinkedIn. Partially doxing each one to obtain Gmail accounts, I found Bob.

Performing database lookups in hope for password reuse (or rules to be applied to their previous password) in order to login with valid credentials.

The way I got into the GitHub was due to Bob, who reused the password "BobsTiger66" (Which GitHub had told was insecure with a red banner, yet he chose to ignore it), and was reused on multiple private databases and one public database (ArmorGames).

Once logged in I had to act quick to avoid GitHub's new ML algorithm to lock accounts out of using new IPs, so I immediately used ssh-keygen to add a new public SSH key to the user profile, I had realised you had added Okta SSO preventing the clone of private repos, in order to bypass this I looked at potential integrations.

CircleCI is a popular CI/CD tool which is inherently linked to organisations via either SSH key linkage or PAT's, therefore I realised the build processes could be exploited in order to obtain private repos, however this was not needed. You guys had added a weird implementation of Okta STS with AWS, producing time-limited tokens, I realised these were spawned everytime a new build process was triggered, therefore I accessed Circle debug mode and managed to extract these time limited tokens, and used them to download your internal datalakes.

Unfortunately these tokens were not given any further privileges, therefore you got lucky or else I would have gained access to RDS via CLI and cloned a snapshot.

When I read this, I was immediately impressed by the level of effort he put into the hack. And despite the outcome, the client was, too.

In the end, this breach had a happy ending, because I was able to provide useful intel to the customer that allowed them to identify how the breach happened, and to also put in proper safeguards to ensure that this did not happen again.

That's ultimately the point, right?

Not to provide customers with a useless writeup of generic TTPs (tactics, techniques, and procedures) regarding assumed threat actors—which is what so many threat intelligence companies do—but to actually provide useful context for how a threat actor breaches their systems.

So many companies just rely on providing existing reports on threat actor groups and never actually get to the core of how an attack happened. Sometimes it takes actually hunting down the threat actors and speaking to them directly. They are usually pretty open and willing to brag about how they did it, because on some level all hackers want to be famous; and as we will see in future chapters, *vanity always trumps OPSEC* (operational security).

In this particular case, I was already speaking to NSFW about several other hacks he is associated with, so it was no issue to ask how he was able to pull this off.

And if you are paying close attention, you will have noticed several misspellings and important "tells" associated with his writeup. Common misspellings or even regional differences in spelling (e.g., organisation vs. organization) can be very important investigative clues that we will discuss in future chapters.

But before we dive into all that, I feel it is important to shed some light on who I am, so you can get to know me a little better, understand what makes me tick, and maybe get accustomed to some of the dry humor and sarcasm that you will find sprinkled throughout this book.

My Story

When I started writing this book, I asked myself a simple question: Am I qualified to write this book? To this day, my answer is still “probably not.” I don’t believe one person can know everything there is to know about a topic, which is why you will find tips and stories from other industry experts throughout this book.

I admire and respect each of the people that I have asked to contribute to this book. I know their work firsthand, which is why I feel they each bring their own unique perspective that complements and reinforces the topics I will be putting forward.

But before we get to that, here is some insight into who I am and what makes me tick.

History

I was about 10 years old when my dad brought home an IBM PS/2. I had no idea what it was or what it could do, but I was mesmerized. This was before the Windows 3.1 days. I remember turning it on and staring at a DOS prompt and just hacking my way through it. The whole thing was like a giant puzzle, which is probably why it sucked me in.

I am a huge puzzle junkie. The more complex, the better. One of my strengths (and also admittedly a weakness) is that I can be relentless when I am trying to find a solution to a complex problem. Some have referred to this behavior as “obsessive.” I get it, and I acknowledge the behavior.

There are nights where I am still cranking away at 4 a.m. because I just can’t stop. It’s part of who I am, and it is a big part of why I feel that I am very good at what I do—whether that be trying to hack into a system or assembling the story behind a criminal investigation.

Roots and Raves

In case you are wondering, I started out my career as a web developer writing HTML and JavaScript in the late ’90s. I grew up in New Jersey and was always into electronic music. Naturally, I was also entranced with the rave culture. Nightclubs like Limelight and Tunnel were the big thing, and I wanted to be a part of it.

Unfortunately these clubs had a 21+ age requirement, which was a problem because I was 16. So I taught myself HTML and offered to build a free website for one of the club’s resident DJs. From then on, I could just walk in with him because I was his “web guy.” Problem solved.

Penetration testing is not much different, which is why I have been doing it (in one form or another) all my life. It's all a matter of understanding what the rules are, then figuring out a way to circumvent them.

I have always been good at finding ways to get around the rules, which I think is a trait shared by most penetration testers.

Don't get me wrong, rules are important. Some people like living their lives in a well-defined sandbox, while others enjoy the challenge of trying to find ways to break out of it. I am the latter.

Developing a Business Model (with Lasers!!)

One evening circa 2011, I was browsing the Internet the same way most people do: with Burp suite active and running passive recon on all sites that I visited.

I was telling my wife about an awesome site that sells high-powered lasers in different colors, hoping she would let me buy one. That was a hard no, but much to my surprise, Burp suite found a passive SQL injection vulnerability in the site.

I had to check it out, and before I knew it, I was able to see the site's user accounts with hashed passwords. Logging in to the site with one of the admin accounts meant having to crack the admin's password hash, which wasn't difficult using any number of online hash crackers given that the password was some variation of Admin123.

I logged in to the site and voilà! I had full access to everything. System records, user accounts, order information, and all.

NOTE Yes, I now realize this action was not exactly "legal," but don't judge. We all have to start somewhere. Plus, this story has a happy ending.

It was that exact moment that I felt the entrepreneurial spark. What if I could take this information and give it to the site's owners so they could fix the injection bug, preventing others from accessing the site in the same way? Surely they would repay this random act of kindness with some of their badass high-power lasers?

I am now the proud owner of a 2,000mW blue laser, and a 1,000mW green laser! Nice, right? The lasers actually burn stuff. They are pretty bad-ass.

More importantly, the site closed the SQL injection vulnerability, and I had a model for a business to provide services that could actually help people.

In the process, I also learned an extremely valuable lesson: If you hack into a website first, *then* try to offer the solution to the customer and ask for a "tip" in the form of a product from their website, it could be interpreted as extortion.

Oops. That clearly wasn't my intent, which I think came off in my email with the CEO, but looking back, I am sure I could have been in some trouble. So while this particular exercise worked out well for everyone, I clearly had to do some work in refining the business model.

Education

One day while I was working for the Department of Defense, I heard from a senior leader that he was going to be bringing someone onto his team that recently completed his “ethical hacking” certification.

Certification? I bet I could do that, seeing as how I already had the skills to hack into things and had been doing it all my life. It sounded like a great career path doing something that I really enjoyed, so I started looking into it.

By this point, I had already earned a bachelor’s degree. I started working a tech-support job while I was in high school, then only took a semester of college before dropping out. It was not until much later that I decided to go back and finish my online bachelor’s degree.

After some research, I found a master’s program at Western Governor’s University (WGU) that specialized in information security and included the Certified Ethical Hacker (CEH) and Certificated Hacking Forensic Investigator (CHFI) certifications as part of the coursework. So I decided to get my master’s degree.

After a few years, I finished my master’s and had all of the certifications that I wanted. Thinking back, I guess I felt a lot like Forrest Gump when he was running across the country: I had already made it this far, so I might as well keep going, right? So I decided to skip the customary CISSP certification and went for my PhD.

I spent about four more years taking online classes and wrote my dissertation on the “perceived effectiveness of the cybersecurity framework among CISOs of varying industries.” I received my PhD in 2018.

Starting Night Lion Security

Having worked with a number of large organizations, including being director of security services for RSM (a top-five accounting firm), I felt that I had a unique perspective on how other organizations performed penetration testing and risk assessments, and I knew I could provide something better.

In 2014, I decided to start Night Lion Security, my own security consulting firm. My vision was (and still is) to assemble an elite force of hackers and penetration testers in order to deliver a report that is thorough and useful.

Being a startup security consulting firm is difficult enough on its own. Being a security startup and trying to compete against giants like Optiv, KPMG, SecureWorks, and AT&T has been brutally difficult.

I feel that I was able to stand out in such an oversaturated market by being heavily active in the news and media. I feel that being on TV is one of the core reasons why companies were willing to take their chance with a small startup that no one had ever heard of. I don’t think I would have been able to make it this far without that.

I have been criticized for this approach because I am seen as the person that is “self-promoting” by going on TV. But in the end, I feel it was worth it because doing so has allowed me to give back in a way that I would not have been able to otherwise. The following is a perfect example.

We recently completed a penetration test for a large, publicly traded bank. At the end of the test, the VP of security went out of his way to tell me that our test was “the first *actual* penetration test they ever had.” All of the big “board approved” companies they used in the past did nothing more than provide a glorified vulnerability scan, and we were able to give them something much more valuable. I am extremely proud of this, and so thankful that he told me because this is *exactly* the vision I set out to accomplish when I started Night Lion.

Digital Investigations and Data Breaches

The transition to digital investigations was so seamless that it wasn’t a transition at all. I wouldn’t even say I “moved” to digital investigations, because it was always just something that I did. Working incident response cases, penetration testing, solving complex problems—it is all the same. It’s all about cracking a puzzle.

As I quickly found out, working on your own cases (i.e., looking for exposed data leaks and breaches) can quickly become more of a challenge in dealing with the aftermath than actually finding and exposing the data.

I uncovered a number of high-profile data leaks including Exactis, Apollo.io, and Verifications.io (which I will discuss more in Chapter 14), and in each case the aftermath of the exposure was different every time.

Verifications.io was particularly interesting because that led to a situation of discovering that the exposed data had actually been stolen from someone else. The company turned out to be completely fake, and once I started poking around, they shut it all down and went running.

There have also been *many* times where I have gone in circles sending copies of data to dozens of companies trying to find the owner.

Something to consider: If you contact a company inquiring about a possible data breach (or leak), that company *is under no obligation to tell you whether the data actually belongs to them*.

Despite the fact that people in this industry may be trying to do the right thing, there are significant repercussions that go along with a company having to publicly admit to a data breach (or leak)—for one, someone is almost always going to get fired, or worse. . . .

In Chapter 14 I detail my own account of the Exactis breach and other discoveries. Let’s just say it’s never fun (or easy) when a CEO sends you a text message on a Saturday night asking why you’ve ruined his life. Here is Troy Hunt, owner of HaveIBeenPwned, with a similar story:

EXPERT TIP: TROY HUNT

An incident that comes to mind is when V-Tech, the Hong Kong toy maker, was breached. This would have been around 2015. This was a huge amount of data relating to kids, including the kids' photos. V-Tech had SQL injection all over the place, it was just an absolute train wreck.

The trouble there as well is it's a Hong Kong toy maker, and as soon as you seem to get to that part of the world, it can be really, really hard to get ownership for these incidents because the company will just sort of shut the doors on you and ignore it, which is what happened in my case.

Breaches with that level of sensitivity are, I think, particularly interesting.

Along those lines as well, the Red Cross Blood Service in Australia had a similar incident a couple years ago insofar as it was a large amount of very sensitive data, including mine. My blood donation application was in there.

This was about half a million Australians, including your blood type and including eligibility criteria or the questions to eligibility criteria such as have you had at-risk sexual activity. It is a perfectly valid question to ask someone about to donate blood, but not a perfectly valid thing to back up from a production server to a publicly facing test server with enabled directory browsing.

The difference with the Red Cross is they just did an enormously good job of their handling of the incident once it actually went live. They regularly stand out now as the gold standard for post-breach incident handling, which is good.

Everything has its ups and downs, but at the end of the day, I love what I do.

This book is the culmination of the past twenty years of my life. I have filled it with real-life stories, scenarios, and techniques that will hopefully one day help you in your own investigations.

With that, let's rock and roll.

Hunting Cyber Criminals

Getting Started

This chapter covers the important items that you should know before getting started, as well as topics like what you will and won't find in this book, the top takeaways from this book that will be discussed regularly in subsequent chapters, and some prerequisites to help ease your journey in cyber investigations.

Some of you may be looking for a reason to get into the field. Some of you may already be in the field and looking for new techniques to use during your own investigations.

In either case, I feel the need to warn you that starting an investigation can be like running a marathon. It can be slow and tedious, and take forever to get where you're going.

You need to be extremely self-motivated because trying to connect dots in an entire Internet of unorganized clues and information can be extremely discouraging.

But if you press on, and muster through that initial pain, it will eventually happen.

There is a feeling you will eventually find during an investigation. It's the same feeling experienced by coders or hackers—it triggers the moment you pull on that first major thread or unlock that first tumbler, which gives way to the second, and the third . . . and eventually the entire world lights up.

There is nothing better or more exhilarating than entering "the zone." It's like a precision laser-focused state—your own "bullet time"—where you can't be slowed or stopped until you've solved the puzzle, hacked the system, or accom-

plished the thing that you're working on. It's a rush better than any stimulant or drug—in a word, it's *amazing*.

Throughout this book, I will provide you with information on my own personal arsenal of tools that I hope will help guide you to exactly that place. I will also provide you with my own experiences and thought processes using many of those tools, because I've found that it can be much more helpful to learn *how* a person uses a particular tool, rather than just re-creating a user's manual.

Why This Book Is Different

I have read a number of digital investigation books, and they all seem to just list every tool possible, provide a short summary of what that tool does, and move on to the next. Almost like herding software cattle.

Many of the OSINT and investigative books I read or referenced before starting this book made me feel overwhelmed with information, like trying to understand a technical encyclopedia without actually giving you any guidance or useful advice tied to what you are reading.

I feel this book is different because I deep dive into the tools and try to provide stories behind actual investigations and how those tools were used in a way that actually proved useful (or not).

Another difference is that the examples won't only show you positive results with every example. I hate when other books do that because the results are typically unrealistic. Real testing often yields no useful information, which is something I will show when comparing different tools.

What You Will and Won't Find in This Book

This book will cover a lot of tools and technical uses of those tools. It will also cover my thought process and the stories behind how I used certain tools to further an investigation.

This book will contain a number of my personal experiences during actual investigations or breach scenarios. While the names may be changed to protect the companies or people involved (but mostly to protect me), the stories and scenarios presented are completely nonfiction. I have a very "out-of-the-box" approach to life, so I will offer life lessons and hacks along the way that may someday help you.

I also don't like that most technical books only feature the perspective of a single person (the author).

I will be the first to admit that I don't know everything about OSINT or digital investigations. Many different facets of technology can come up during an investigation that may require a unique perspective or an understanding that comes from years of hands-on experience, which is why I have always tried to surround myself with people that I respect and that I feel are experts I can learn from.

I thought it would be really interesting to you, the reader, if I also included the opinions and experiences of some of those people alongside my own. Since I am writing a book on a subject, why not also include the opinions of people who are also really good at said subject?

So I asked a handful of people that I consider experts in their field to contribute a story, an opinion, or even a technique on some part of the information-gathering or investigative process.

I found each of their stories to be unique and thought-provoking, and I know you will, too!

Getting to Know Your Fellow Experts

I would like to give a very special thank-you and shout-out to the following people for their contributions as experts in this book (in alphabetical order):

- Alex Heid
VP research, SecurityScoreCard & founder of HackMiami
- Bob Diachenko
Security Researcher, Founder of SecurityDiscovery.com
- Cat Murdock
Threat and Attack Simulation, Guidepoint Security
- Chris Hadnagy
Chief Human Hacker, Social-Engineer, LLC, SEVillage owner
- Chris Roberts
Chief Security Strategist, Attivo Networks
- Leslie Carhart
Principal Threat Hunter, Dragos, Inc.
- John Strand
Founder, Black Hills Information Security, Senior SANS Instructor
- Jonathan Cran
Founder, Intrigue.io, Head of Research, Kenna Security
- Nick Furneux
Computer Forensic Investigator, Crypto Investigation Expert
- Rob Fuller
Red Team Heavyweight
- Troy Hunt
Security Researcher, Microsoft VP, Founder, Have I Been Pwned
- William Martin
Researcher, developer of SMBetray

A Note on Cryptocurrencies

An extra super shout-out to Nick Furneux for writing the primer to crypto investigations later in this chapter. For those interested in really diving into how to investigate cryptocurrencies, please check out his book, *Investigating Cryptocurrencies: Understanding, Extracting, and Analyzing Blockchain Evidence* (published by Wiley).

What You Need to Know

The following themes will be discussed regularly throughout this book and should be considered key takeaways. The takeaways are ordered by a natural flow of information and not by order of importance:

1. When dealing with a young and aspiring hacker (aka skid aka script kiddie), *Vanity will always trump OPSEC*. This book will provide many examples proving this statement.
2. Access to historical information can often make or break an entire investigation. If a young or aspiring cyber criminal is willing to sacrifice OPSEC for their own vanity, then being able to look back in time will most likely lead you to the answer you are looking for. The more historical information you can access, the better the odds of finding whoever or whatever it is you are looking for.
3. You always get what you pay for. If you want access to the best and most complete sources of historical information, it won't be cheap. If you want cheap (or free), don't expect to have access to everything.
4. Never rely on one tool for all of your answers. You should always try all tools and techniques at your disposal, even if in the past they have not provided any useful results. Sometimes you get lucky. This book is full of examples where I was completely blown away by the results, which is why you should. . .
5. *Save everything*, and keep meticulous documentation so you can find it later.

WARNING This is the worst part of any investigation, especially when you are on a roll finding new details . . . so I can't stress this enough. There are a few key items from my own research that I can't believe I did not save. I was sure I took screenshots, but I must have been so consumed with the research (and in the zone) that I forgot to save the items. Now those items are gone and I kick myself every day about it.

So again, *save everything*.

Paid Tools and Historical Data

Throughout this book, I will do my best to use free and open-source tools as much as possible, but back to takeaway #3, you *always* get what you pay for.

Intelligence research and information gathering is no different. You can always go the cheap route and depend on truly “free” tools, but it may end up costing you in the quality of information that you are able to retrieve and the amount of time you spend looking for the information.

I truly believe that the most crucial part of an investigation will often come down to the level of historical data you can access. It will be rare to find open-source tools with much of a backlog of historical data.

A few tools in particular contain a wealth of information that I gladly pay for. I will talk about those tools in greater detail later, but for now, just know that not all the techniques I discuss will use completely free tools. It’s a trade-off. You will need to make a decision on whether you want to spend money—but just be aware that not everything can be free.

What about Maltego?

Maltego is a powerhouse tool for digital investigations, perhaps even *the* industry standard investigative tool. It has been covered extensively in just about every other digital investigations book, and certainly in books dedicated to only covering its many uses and applications, which is why I made the decision to leave Maltego out of this book.

Don’t get me wrong, I use Maltego religiously, but the program is so vast that in order to cover it properly, I would have to dedicate most of this book to that one topic. So many other useful and noteworthy tools are available that just don’t get the attention they deserve. Now they will.

Prerequisites

Only two prerequisites are required to effectively use the tools and techniques described in this book.

Know How to Use and Configure Linux

The majority of the tools and examples provided will be in Linux. Having at least a basic understanding of how to run the commands will be important.

It will be up to you to set up and install the tools and their respective dependencies. You have many different Linux environments to choose from, each with its own set of benefits. I would rather spend time focusing on techniques and stories to help in your investigation, instead of trying to provide exhaustive tech support.

If you're not sure how to set up your own Linux distribution, I highly recommend downloading Kali Linux. The majority of everything you will need will already be set up for you. You can download Kali at www.kali.org.

Get Your API Keys in Order

Many of the tools in this book will have API connections to multiple sites/services. One of the most frustrating things to deal with during your initial setup is having to set up the API keys in each tool.

Keep a master list. There is not much more I can say about this topic, but I want to call it out because of how much time it will save you in the long run. If you don't have a long list of API keys, that's OK. Start with just one. I use the format Sitename: APIKEY, and I store everything in my 1password vault. It's simple and easy to get to when I need it.

I would probably stay away from posting your keys on any public site like an AWS bucket, a Trello board, or a OneNote file. I never knew Trello boards were publicly searchable until one day there was a story about passwords and other account details being exposed on the service. My point is that I would probably steer clear of posting your keys or passwords on something that you don't have direct control over.

Important Resources

The following resources are extremely useful guides to help advance your knowledge of OSINT and investigations.

OSINT Framework

The OSINT framework is a collection of Open Source Intelligence tools designed to make the process of gathering intelligence and data collection easier.

The OSINT framework provides an exhaustive list of tools (much more than what is covered in this book) in an easy-to-use web interface.

The online interface (shown in Figure 1.1) provides categories and classifications for different intelligence sources, making it an important checklist (or road map) to ensure you are investigating all possible sources.

The OSINT framework is an excellent resource for investigators and penetration testers, and you can find it at <https://osintframework.com>.

OSINT.link

OSINT.link (www.osint.link) provides an exhaustive resource of links, search engines, and web directories designed to help gather information. Figure 1.2 shows the different "search engines" categories, which is only one of many available parent categories of available links and resources.

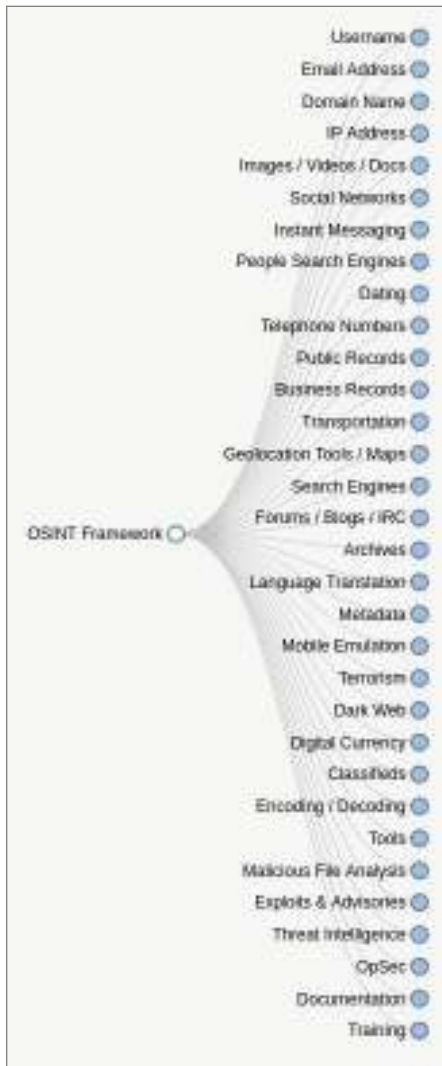


Figure 1.1

IntelTechniques

Fellow author and OSINT expert Michael Bazzell hosts www.inteltechniques.com, which until recently, provided very useful social media and investigative search engines. As of April 2019, this site is no longer free and only available to paid members of the site's video training. See what I mean about you get what you pay for?

I have used this site on many occasions, and therefore it's worth suggesting if you are willing to pay for the training.



Figure 1.2

Termbin

I love Termbin (www.termbin.com).

Have you ever been in a situation where you are working in Linux and need to send yourself some text data but can't directly transfer the data out of the machine?

Rsync may not be an option if you are connected through one or more levels of jump servers. Setting up a web server would mean punching a hole through the firewall, which is probably a bad thing, so what do you do if you need to send yourself text from one of these servers?

In this case, Termbin is for you!

Send the contents of the file to `termbin.com` using `netcat`, and receive a private link where you can download your text.

For example, let's send a file called `surprise.txt` to Termbin using the following command:

```
root@osint > cat surprise.txt | nc termbin.com 9999
```

You will get back a custom URL that looks like this: <https://termbin.com/cpc4>.

Hunchly

Hunchly is a last-minute addition to this book. If I had known about the tool earlier I would have covered it in greater detail in another chapter because it is utterly fantastic.

Hunchly is a tool for online investigations that automatically collects documents and annotates every web page you visit.

There are so many instances where I wish I would have saved a particular screenshot, or even when I know I saved it, but I just can't find it. Hunchly eliminates that by capturing everything in your browser and tagging it to a particular investigation.

If you are an investigative professional, or even just an OSINT enthusiast, **you absolutely should download this tool.**

Hunchly is available for Windows, Mac, and Linux, and directly integrates with Chrome for a seamless experience. When you are ready to start researching a particular investigation, you can quickly switch on the plugin, select which case you are working on, and Hunchly will do the rest for you.

I wish I could have covered this tool in greater detail because being able to properly save and document your investigation findings is so incredibly important, even from the standpoint of being able to go back and figure out how you reached a particular conclusion.

Hunchly is available at www.hunch.ly.

Wordlists and Generators

The use of wordlists will come up regularly in this book. Plenty of generic wordlists are available, and those will typically get the job done in most situations. When you are bruteforcing or looking for hidden treasure you will typically want to use your own list variations.

SecLists

Your first stop should be the SecLists GitHub page.

SecLists is a GitHub page maintained by Daniel Miessler and is home to an excellent collection of many different types of wordlists including usernames, passwords, name combinations, data patterns, fuzzing payloads, and many more.

Download the wordlists on the SecLists GitHub page at <https://github.com/danielmiessler/SecLists>.

SecLists provides a great starting point for wordlists. You can use the following tools to create your own custom wordlists, all of which are available in Kali Linux.

Cewl

Cewl is an open-source custom wordlist generator designed to build wordlists specific to your targets. Cewl builds its lists by spidering target URLs and returning lists of keywords that can be used in various password-cracking apps like JTR.

If you are looking to build a list of keywords specific to your target, Cewl is the sniper rifle you are looking for.

Think of how many specific keywords you can gain by scraping a person's social media pages. Once you have those keywords, you can generate permutations using the app's standard combinations or your own custom syntax.

You can download Cewl at <https://github.com/digininja/CeWL>.

Crunch

Crunch is a free tool that generates complex and exhaustive wordlists using custom patterns and permutations.

Crunch would be your shotgun approach to developing a wordlist, and an excellent tool to use when looking for obscurely named public repositories and S3 buckets.

You can download Crunch at <https://sourceforge.net/projects/crunch-wordlist>.

Proxies

When running OSINT searches (such as NMAP scans or directory bruteforcing), it may make sense to use proxies to avoid detection.

One option would be to purchase 50 or 100 private proxies that you can auto-rotate through a tool like ProxyChains. Hundreds of proxy sites are available where you can purchase private, high-quality proxies.

My two favorites are:

- Lime Proxies (www.limeproxies.com)
- Squid Proxies (www.squidproxies.com)

Storm Proxies (Auto-Rotating)

My favorite proxy site for OSINT searching, investigations, and web scraping is Storm Proxies (www.stormproxies.com).

Storm Proxies automatically rotates proxies for you without the need to set up ProxyChains or some other rotating proxy service on your server. With Storm Proxies, you send all requests to a specific IP address, which then routes your traffic through one of thousands of its own private proxy servers.

You can choose to use a 3- or 15-minute proxy, which changes your IP every 3 or 15 minutes, respectively, or even send every request through a different proxy server (which is an excellent way to avoid firewall detection).

Storm Proxies is a paid service, but fairly inexpensive considering the amount it would cost to purchase and manage hundreds of private proxies yourself.

Now a word from Nick Furneux: an introduction to investigating crypto currencies.

Cryptocurrencies 101

By Nick Furneux

At the end of 2008 the enigmatic Satoshi Nakamoto wrote a whitepaper about a self-creating, self-managing currency based on a new type of database called a blockchain. In early 2009 a proof-of-concept blockchain system called Bitcoin was created, which promised to revolutionize currencies with its model of decentralization—essentially, no banking or government control. Bitcoin has gone on to be something of an enigma itself, not really fitting the criteria of a currency while somehow generating an accepted value and tradability. Most commentators now prefer the term *cryptoasset* rather than a *cryptocurrency*.

The blockchain concept is essentially a clever way of storing contracts such as coin transactions in a database that protects its data using cryptographic methods and makes it very difficult for an attacker to change entries in the database without significant processing power. The term *blockchain* has joined other technology terms such as *AI* and *Cloud* to be used to sell systems that rarely require the stated technologies to function well, or indeed to improve current methods, but they sound good on marketing material and hence people buy into them. In 2017 a company called Biopix changed its name to Riot Blockchain and saw its shares grow by 394%, a trend that dramatically reversed after CNBC broadcast the suggestion that the name change had been done purely to boost the company's value!

Since 2009 many new cryptocurrencies have been built; at the time of writing, www.coinmarketcap.com lists 2,164 tradable cryptocurrencies of different types. Most use a variant of blockchain; however, we have started to see different technologies such as the “Tangle” used by IOTA. Some cryptocurrencies provide partial anonymity of users and some are very anonymous indeed. The benefits are all very similar and include:

- No central controlling authority
- No clearing of funds, leading to faster transactions
- An immutable transaction history
- Partial or complete anonymity

As most popular cryptocurrencies have open-source ledgers of transactions, they fit within the definition of open source and the public ledgers can be very useful to an investigator. However, the pseudo-anonymous data used by cryptocurrencies such as Bitcoin and Ethereum can make it very difficult for an investigator to make sense of the data.

How Do Cryptocurrencies Work?

Cryptocurrencies are fairly complex systems, and entire books have been written about their internal workings, which we will not try to duplicate here. However, a few details will be useful to the OSI investigator.

A cryptocurrency address is a combination of letters and numbers, which is often represented in Base58. They tend to be 34, 42, or even 96 characters long, although there are other lengths.

A Bitcoin address looks like this: 1BoatSLRHtKNgkdXEobR76b53LETtpyT

An Ethereum address looks like this: 0x89205A3A3b2A69De6Dbf7f01E-D13B2108B2c43e7

A Monero address looks like this: 44AFFq5kSiGBoZ4NMDwYtN18obc8Aem-S33DBLWs3H7otXft3XjrpDtQGv7SqSsaBYBb98uNbr2VBBEt7f2wfn3RVGQBEP3A

An address is essentially a public key that a user of the currency can give to anyone who wishes to send them coins. However, to move any coins stored in an address the accompanying private key is needed. (You can find an excellent overview of public/private key systems here: <https://medium.com/@vrypan/explaining-public-key-cryptography-to-non-geeks-f0994b3c2d5>).

When researching online the investigator will see many cryptocurrency addresses, but these will always be the public key address. Although some private keys end up online as either leaks or mistakes by users, this is very rare and any coins controlled by the key will usually be long gone.

Cryptocurrency addresses are pseudo-anonymous in that they are not directly linked to an account or identity, but techniques exist to be able to infer ownership even if this is complex and time-consuming to achieve.

When a user wishes to send a cryptocurrency value to another user, they construct a transaction that is communicated to every ledger (full-node user) of that currency. So-called miners then add transactions to a block (think of a mental picture of a box full of transactions) and use super-powerful computers to solve a complex mathematical problem that locks the block so that data cannot easily be changed. If anyone tried to change the elements of a transaction, they would need to communicate that change with every ledger on the planet and recalculate the mathematical problem. The more blocks there are to calculate above the block containing the changed transaction, the more difficult, or indeed impossible, it is.

Blockchain Explorers

With the exception of some closed or very secure cryptocurrencies, the ledger is open source and therefore a useful resource to an investigator. If an investigator locates an address that belongs to a suspect, such as in their forum post signatures, for example, it is simple to ascertain the value that has passed through the address or perhaps is still being stored at the address. To do this we need to use a block explorer. You can find many examples for all the major currencies.

For Bitcoin, some examples are:

- www.blockchain.com
- www.blockcypher.com
- www.btc.com

TIP When using blockchain.com, search for an address and then scroll to the bottom of the page and click the Advanced – Enable link. This will provide values for each input and output.

Others, such as www.blockchair.com and www.bitinfocharts.com, provide explorers for many currencies such as Bitcoin, Bitcoin Cash, Dogecoin, Ethereum, Litecoin, and more. They all essentially display the same information but in different graphical interfaces, and so it's important to understand the primary elements.

Let's take a look at blockchain.com/explorer. Here we can search for a Bitcoin, Bitcoin Cash, or Ethereum address. This will return a list of all the transactions where the address has been an “input,” where it paid money to another address, or an “output,” where it received monies from another address.

If we take a look at a Bitcoin address we see a block of metadata in the top panel followed by a series of transactions (Figure 1.3).



Figure 1.3

The metadata about an address can be very interesting as it can help us build up a picture of the use of this address. For example, how many transactions has this address been involved in? How much Bitcoin has been received by this address over time and how much is left there now? The values also have a graph option. This will show the movement of coins over time. When was the first payment made and what patterns do we see? Are coins received then immediately moved on or retained in the address? When was the first payment made?

Why are these questions important? Let's take the example of a scam where a victim's computer has been infected with a virus that encrypts all the data, and a Bitcoin ransom is demanded for an unlock code. The questions posed could tell us when the scam likely started, how many victims paid into the scammer's address, how much money has been made, whether the coins are moved on or retained, and so on. In the case of the Petya/NoPetya ransomware, the coins were retained in the address for several days after victims paid before moving the coins away. This can clearly be seen by searching for the address used by the scammers: 1Mz7153HMuxXTuR2R1t78mGSdzaAtNbBWx. By taking a look at the graph for the balance and highlighting the obvious peak in June 2017, it is straightforward to ascertain when victims started paying, how much was received, and when the coins were moved.

It is clear from the graph that payments started on June 27, 2017, and primary payments were finished by June 28, 2017. Coins stayed in the address until July 4, 2017, when they were transacted. This helps us to understand the life cycle of the scam; victim payments really took place over a 24-hour period and were retained in the address for just a short few days before they were moved. No other significant payments were made. This is a good example of how metadata of just a single address can help us to build a picture of an address used in a crime.

The number of payments may help us to discern the likely number of victims, and a site called <https://oxt.me> helpfully breaks this down for us. Browse to the site and look at the address 1Mz7153HMuxXTuR2R1t78mGSdzaAtNbBWx again (Figure 1.4).



ACTIVITY	
FIRST SEEN	JUNE 27, 2017
LAST SEEN	MAY 7, 2019
INCOMING TXS	89
OUTGOING TXS	4
VOLUMES	
RECEIVED	4.42475836 B
SENT	4.26445825 B
BALANCE	0.16030011 B

Figure 1.4

I find this data very helpful as it quickly breaks down for us when an address started to be used, when it was last seen, and separate values for the coins received and sent. The breakdown of incoming and outgoing transactions also helps us understand, in this example, the number of likely victims, 89 incoming payments, and how many addresses were used to eventually move the coins away from the scam address. If we were analyzing payments into an illegal dark-web store, for example, then the incoming payments would likely be customers.

TIP Victims of a scam are usually cryptocurrency novices and hence tend to purchase coins straight from an exchange and send them to the scammer. If the exchange can be identified, it is easy to make a request to the exchange for KYC (Know Your Customer) information and contact victims.

Following the Money

When investigators describe cryptocurrency investigations, they often mean following some type of illicit payments from a source address to a destination. Because of the pseudo-anonymous nature of cryptocurrencies, such as Bitcoin with its 34-character Base58 addressing, this is complex to achieve. What I describe as “address blindness” can quickly set in as you click from transaction to transaction.

To move from transaction to transaction is slightly different in all the various blockchain explorers. Using Blockchain.info you use the highlighted “Spent” link on the output side of the transaction to move to the next/later transaction, and rather confusingly click the “Output” link on the input side of the transaction to move back to the previous transaction (Figure 1.5).



Figure 1.5

Let's assume for a moment that you have received a report that the Bitcoin address in Figure 1.5 (38fUXUxwunFJcZtzfSYh21z9Zu1EXGX2np) is a scam address. It has 11.964 . . . Bitcoin in it. Where does the money go? We can see that 11.66 goes to address 3Gpor . . . and a small amount to 153b . . .

NOTE I only use the first five characters when annotating Bitcoin addresses. The patterns stick in your memory much more effectively than 34 characters and help with address blindness. Don't worry that you will see another address with the same five characters; that works out as (forgetting the first character being 1 or 3) $34 * 34 * 34 * 34$, which is over 1.3 million permutations!

We next need to ascertain if one of the addresses is a payment address. In the vast majority of transactions that are one-to-many or many-to-many addresses, one of the outputs will be a change address. The reason is that each Bitcoin is made up of 100 million satoshis, so the chances that payments to several addresses will add up to exactly the input amounts is tiny. In this example, the payment is from a “3” wallet. Bitcoin addresses that start with the number 3 are used for multi-signature or more complex transactions than just a straight address-to-address payment. The outputs are to a 3 address and to a 1 address. It is significantly more likely that the change address is also a 3 address. Hence we can infer that the 3 address is the change and the 1 address is the payment.

Other techniques exist for discerning which output address is the change address. For example, in Figure 1.6 any of the 3 inputs would have paid for the 3HfFw . . . payment, but all 3 are needed for the amount sent to 3B5kC. This helps us conclude that the change address is the 3HfFw address and hence we need to follow the other one.



Figure 1.6

Once we have discerned the address or addresses that are likely payments, we can follow them to the next transaction by clicking the Spent button. In this example, both the outputs are Unspent so we are at a dead end.

Clicking to the next or previous transaction will just provide another list of inputs and outputs, and the investigator can get lost immediately. I recommend stopping after each click and researching the addresses, and build the picture again before moving on. Once you have moved two to three transactions from your start point, experience shows that you are often beyond the scope of the investigation. Unless your suspect is using mixers and tumblers to move and split the coins, which is extremely hard to follow without the help of a commercial tool, then payments of interest tend to be within just a few transactions of your start point.

I have worked on numerous transactions where this is true. There is even a counter terrorism investigation where people donating to the “cause” were just buying from exchanges and donating straight to the group, and then the monies were cashed out in just one or two hops to another exchange. You could draw the entire transaction graph on a small piece of paper.

Of course, there are also times where criminals are extremely sophisticated and will go to extraordinary lengths to obfuscate their eventual destination through

splitting and recombining coins, moving through exchanges, or swapping into other currencies. This is where commercial tools really come into their own.

Identifying Exchanges and Traders

Moving from transaction to transaction is straightforward but fundamentally meaningless unless we can identify a trusted and real-world resource that may have KYC information or log data such as IP addresses. Identifying these resources is key in most investigations. It should be noted that it is not possible to track through an exchange. A deposit of a coin to an exchange followed by a withdrawal will not return to you the same coin any more than paying \$10 into a bank and then withdrawing it from an ATM will return the same note. Once a deposit has been made, your only chance at tracing is the exchange working with you to supply the required KYC data.

Several websites attempt to identify the addresses belonging to exchanges and traders; the most reliable are www.walletexplorer.com and www.bitcoinwhoswho.com. The latter site also has a scam reporting function and also searches the web looking for mentions of the addresses you searched for.

Walletexplorer.com attempts to use clustering techniques to identify one address and then infer ownership of other addresses owned by the same entity. This is the same method used by the commercial software offerings but without the budget! The process is simple: make a deposit into an exchange and they will provide a payment address; withdraw a coin and you will see the input address. You now have two addresses you know are owned by the exchange.

Next, use clustering techniques and algorithms to locate other addresses owned by the same entity. A full discussion of clustering is beyond this introduction, but as an example, look for transactions where the known address and the input address are shared with other input addresses. It is likely that those addresses are owned by the same entity.

This is where the commercial tools excel, and if you are serious about investigating crimes involving cryptocurrency, your life will be very hard indeed without purchasing one of the primary tools. All of the tools attempt to identify clusters of addresses owned by exchanges, traders, darkweb sites, and others to assist the investigator. Most have visualization capabilities to make the job of tracing from your suspect transaction to an exchange or other known entity much easier.

At the time of writing the primary commercial tools are:

- Chainalysis (chainalysis.com)
- Ciphertrace (ciphertrace.com)
- Elliptic (www.elliptic.co)
- Coinfirm (coinfirm.com)

Maltego (not covered in this book) also has free transforms available to install from the Hub. These are provided by blockchain.com and provide transforms to locate input and output addresses from transactions. This can be very useful as a low-cost visualization tool. Ciphertrace also sells transforms to give access to its Bitcoin identification database from Maltego.

Summary

This chapter provided an overview of the type of content and tools that you can expect to find throughout this book. This chapter also provided a handful of technical resources that should be considered staples in your journey through this book and future investigations.

The next chapter will provide background information and stories relating to the threat actor groups like The Dark Overlord and Gnostic Players and will set the groundwork for what it means to investigate cyber criminals.

Investigations and Threat Actors

This chapter will focus on what it means to be a cyber investigator. We will delve into different types of researcher and investigator roles, some moral challenges you may be faced with, and different paths you may have to take in order to complete your research. This chapter will also provide an overview and introduction to the different threat actors and groups that will be discussed throughout this book.

Before we start, I would like to apologize for the literal sea of usernames and aliases that I will be dumping on you throughout this book. Some threat actors use different aliases on each forum or website, while the more experienced actors will intentionally use aliases belonging to other known hackers in the community. Also Known as “Alias-Hijacking.” This can be incredibly frustrating and confusing, which is compounded by the fact that threat actors may even swap aliases within their group just to throw off investigators or law enforcement.

The Path of an Investigator

To quote Drax from *Guardians of the Galaxy 2*, “there are two types of people in this world. Those who dance, and those who do not.”

I feel like that is a pretty accurate summation for a fundamental question in how you conduct your investigations: do you want to get involved with the

people and organizations you are investigating, or do you want to go Stealth mode and quietly observe?

Depending on your job, there is a good chance this answer will be provided to you based on corporate and legal guidelines. Some companies have a strict non-interaction policy. In which case you don't get to decide. In many cases however, there will be some leeway here, so it will be up to you to decide how you plan on collecting your information.

Obviously every style has its own advantages and disadvantages. I have personally never been one to quietly sit back and observe. It is the way my brain is wired—I need to be constantly in motion. So for me, I have to get involved.

Go Big or Go Home

I typically juggle four or five active aliases at any one time. Each alias has a cover story, and some character background. I try to really get immersed in whatever I am doing, so if I am looking at a Russian carder, I may try to have a broken-English accent, and use a name that is somewhat familiar in those circles.

I also will always try to play on another actor's name. People switch aliases and contact information so often that it is very easy to start up a new account with a similar name and try to reach out to people in their network. Alias hijacking has many benefits, and if you are good at socially engineering people, you can use an older alias to quickly elevate yourself in the ranks. If you are convincing, there is a good chance you will get lucky and talk to someone who thinks you are someone they used to know.

But again, it all depends on what your goals are.

EXPERT TIP: CAT MURDOCK

I think the important thing to remember about open-source intelligence, OSINT, is that it is really just *open-source information* until you can make it *actionable*.

For example, if you are gathering intelligence on a corporation, or preparing for a wide-scale phishing campaign, you will be looking for very different information than you would be for a targeted spearfish on one specific individual. You will be looking for more information about the target network and events that might be happening with the company as a whole, rather than events that might be happening to the individual.

So whenever you start your investigation, it is really important to take a step back and ask yourself why you are investigating this target. If you're looking to perform a spearfish, then it kind of opens up a whole different avenue for investigation because you have to care much more about who that actor is, and specifics about the person.

What drives them? Who do they interact with?

Then the next question you would want to ask yourself—within the scope of the engagement—is how can you contact this person? And then, what intelligence do you actually want? Because even though certain information like housing records may be helpful in certain contexts, you won't be using that information to try and spearfish your target.

If it is allowed under the guidelines of your investigation, making direct contact with your target can obviously provide you with information that you might never be able to find online. Direct communications with your targets can be extremely beneficial—and extremely dangerous. If you are not careful, direct contact with a real threat actor or criminal can have very serious consequences for the investigation.

When interacting with threat actors, I really enjoy coming up with cover stories and trying to get directly involved with the groups I am investigating. One “bonus” of working your way into one of these groups is that they are *always* busy. The more you get to know them personally and the more they trust you, the more likely they will share new information with you.

They are always working on some new hack, and once you are in their inner circle, they are typically more than willing to brag about their latest hacks or exploits.

I feel that this kind of hands-on threat intelligence gathering is missing from most organizations. Most threat intel teams I have come across that look for passive information, but very few that I have seen actively go looking for it.

You might be thinking to yourself, “Yeah, this all sounds great and exciting, but how useful is this? Is it really worth the effort?”

I’m glad you asked.

The Breach That Never Happened

One evening, I drafted a letter to the CEO and CISO of a major airline company. The title read “– URGENT – Data breach in your network.” I received a phone call back later that evening, and proceeded to tell the security admin that I received word from one of my darkweb contacts that data from their network was about to go on sale.

Specifically, the data was “about” to go on sale because the hacker was still *in* their network, exfiltrating the data.

They said they would look into it, and after about two days they got back to me. *They could find no evidence that someone was in their network or had breached their systems.*

The hacker was also selling access to the network (a common practice), so I reached back out to the hacker and asked for proof of access before I completed the purchase. He was nice enough to provide me with screenshots of live commands he was running inside the company’s network, which also included a list of the company’s admin accounts.

The following is part of the output I was provided as proof of access to the network:

```
beacon> shell net group "domain admins" /domain
[*] Tasked beacon to run: net group "domain admins" /domain
[+] host called home, sent: 64 bytes
```



```
[+] received output:
The request will be processed at a domain controller for domain ***.com.

Group name      Domain Admins
Comment         Designated administrators of the domain
Members
-----
A****          a****          a****
A****          A****          A****
A****          a****          a****
A****          a****          A****
M****          SP****        svc_****
svc_****
```

The command completed successfully.

What Would You Do?

This scenario presented me with a big dilemma. On one hand, I could take this new information and present it to the company so they could find and lock out the hackers before the exfiltration of their data was finished.

On the other hand, doing so would mean that the hackers would most likely realize that I am the leak, and push me out of their circle. Months of hard work infiltrating their ranks would be thrown away, and my aliases would be blown.

What would you do?

I decided to contact the company again and let them know that I had more information. I also let them know that giving them this information would easily blow my cover, so I would really appreciate it if they would provide me with any intelligence they gathered from their investigation. I already knew how they were able to gain access, so I was specifically after the IPs used in the attack.

They agreed to reciprocate.

This company was not one of my customers, and I had no previous relationship with anyone on their security team. I knew going into this situation that they were under no obligation to provide me with anything after I provided them my end of the information.

I chose to move forward anyway, fighting off my natural inclination of assuming they would not hold up their end of the bargain. I sent the info back to the admin at the airline company, and (not surprisingly) received no response back.

After two more days of unanswered emails, I received a phone call from the admin. He said he appreciated my help—they were able to find the user because of the data I provided, and successfully booted the hacker out of their network.

Unfortunately, legal intervened and he was not even supposed to be talking to me. Because of all the press stories and data breach disclosures associated with my name, the company was afraid I would go to the media with the information.

At least he was nice enough to call me back and tell me that the information was real, but he could not give me any other information.

The organization ultimately never disclosed and decided to ignore the entire situation. You should keep that in mind when deciding what to do with the information you have uncovered.

NOTE Yes, my contact stopped speaking to me. It wasn't hard for him to figure out that I was the reason he was booted out of the network. I have spoken to a number of other threat intelligence companies whose position is very clear on never getting involved, probably for this exact reason.

Sometimes, getting a customer to even admit the data is theirs can be a challenge. Here is Troy Hunt, owner of "Have I Been Pwned" (www.haveibeenpwned.com), with a similar story.

EXPERT TIP: TROY HUNT

A while back there was a guy on Twitter who went by 0x2Taylor. He was just constantly dumping pretty full-on stuff. Just, bam! Full breached data straight on his timeline. A couple years ago he dumped one that had hundreds of thousands of payment records from a company called Blue Snap (Figure 2.1).



Figure 2.1

Full, legit payment records which included full card numbers, CVV, person's home address, phone number, just the works. All in clear text. When I went through and started looking at the data, I saw a lot of references to a company called Regpack. It was a little bit weird because the data was being presented as one thing (Blue Snap), but looking at the data implies another owner.

What I discovered is Regpack runs a registration service that you can embed on your web host. With just a bit of JavaScript, whether you're a scout group or a school charity, you can now take payments on your site.

The quote from Regpack was "We have run the full security protocol implemented in these cases and conclusively determine that our servers were not involved." Then literally the next day, they turn around and issue the following statement that the data did come from them.

Their statement was, "We identified that a human error caused these decrypted files to be exposed to the public facing server, and this was the source of that loss." Then they continue and they say, "Regpack systems were not breached."

But there are 324,000 payment records exposed in plain text. . . . So what happened? If it wasn't a data breach then what was it? They said it was a "data incident." Not a "breach," but a data incident. Oh, come on . . . please.

I kind of wondered if they were trying to avoid the *breach* word because they had some specific regulatory obligation that would have applied if they actually had a breach . . . but you just leaked hundreds of thousands of full account numbers and CVV codes. You have a regulatory requirement with PCI DSS that's going to slap you already.

It's always fun dealing with people that will try to deny the data came from them, despite the overwhelming evidence.

Moral Gray Areas

During the course of an investigation, you will inevitably come to a number of morally gray areas. Whether it's looking for public data leaks, investigating threat actors, or even infiltrating specific threat groups, there is always a line that you may have to cross in order to achieve your goals.

Threat actors and criminals will never respect moral or legal boundaries, so it might be an immediate red flag if they see that you are trying to uphold some moral standard. Something to keep in mind—crime is their livelihood. Cyber criminals are there to make money, so someone legitimately in the scene for the same reasons will inevitably take the conversation towards something illegal. Why are you there, if not to make money? This is typically the first and most obvious red flag, and why so many investigators are discovered immediately.

These scenarios come up quite a bit, especially during social engineering, which by definition means you are lying. I discuss several of my own scenarios in Chapter 15, when we dive more into social engineering. For now, here is a very interesting viewpoint by John Strand.

EXPERT TIP: JOHN STRAND

When SEing a target, there is nothing more powerful than hate. Where, if you start trolling someone, like if they're a very religious person and you come at them and say, "Well, I'm an atheist and everything you believe in is wrong, Jesus was a manufactured idea by the Roman Empire to control the Jews," and really get people riled up.

You get 'em riled up and you can give them links and you can be like, "Here's a link at this website, refutes everything that you believe in." **That person is going to click that link.**

The downside on that is, most of our customers would never, ever, let us go that far because that's pretty nasty territory.

It is definitely crossing a line and I think that if you can talk about it in a book, we do this professionally. We do very evil things with a contract and when you start talking about social engineering, when you start talking about recon, specifically on individuals, you are really starting to get close to a line, where it blurs out of the engagement and it starts blurring into something more personal.

Being cognizant of where that line exists is something that many people in this industry don't have, until they cross it. They don't understand what they did wrong, and I think that that's an important concept. So, you know, that's number one.

Number two, I am a really big fan of recon against evil people. How can you do recon against an adversary? Cyber attribution. There's some tools that I love to use if we're trying to get some level of cyber attribution. Honey badger, a meta tool written by [Tim Tomes 00:11:03] while he was working at the HIS, is an amazing tool where if you can entice an adversary to run an applet, to run a Word document or an Excel spreadsheet with a macro, we can geolocate that person within 20 meters.

If you go to Canary Tokens, they have amazing tokens that you can create in Word documents, PDFs, AWS keys. Set up something where if an adversary clones your website, it'll beacon back and this is also a form of recon. It's a whole form of recon against adversaries and cyber attribution with Word web bugs where many people in the industry don't think about it. A lot of the tools and techniques that we have as red teamers, they can be very effective in the blue team realm, to get a high degree of confidence in cyber attribution against an attacker, if you know how to leverage them properly.

Different Investigative Paths

There are many different career paths you can choose that involve cyber investigations. In an incident response role, you will more than likely focus on investigating the crime more than the perpetrators. If you work in a security operations center (SOC), you may be looking to validate and understand different types of potential threats coming at you from many different forums of data feeds.

Even then, your industry may dictate the type of investigative work you do, or your role could involve both proactive and reactive investigative work.

EXPERT TIP: LESLIE CARHART

I'd say the large lion's share of my time is divided between hunting, which is proactive, and incident response, reactive.

We are very heavily involved in identifying advanced adversaries. Not necessarily attributing them to nation states, but attributing them to organized groups, to understanding which groups are doing which activities, who they're targeting, and how they target them.

We're very cognizant of specific actor groups, and who they're attacking and how they're doing it. We're always trying to gather more intel on who's out there in the industrial space wreaking havoc. That's a major part of our business and our community work as well, is identifying adversary groups and understanding how they operate.

Like any incident response, there is not a lot of satisfaction in seeing people caught for the attacks that they caused. That's not our place as a security firm. That requires international law enforcement cooperation.

But we are always trying to improve our detection and prevention based on what we know about how adversaries function, how they have to function, how they've been functioning. That's a major part of doing good incident response and security operations.

For example, with industrial actors, there are things that the advanced adversaries are doing more on math, like reconnaissance, those things are pretty easy to track because they're being done to a lot of organizations simultaneously. Unique and novel attacks against systems like actually causing kinetic impact to systems, there are some things that are reused, but normally they are so cutting edge and novel that there's a lot of newness.

I basically divide the attacks against the industrial systems that are occurring right now into three categories. The first one would be advanced adversaries. People who are either doing reconnaissance and building footholds inside industrial networks, or actively in some rare cases launching attacks.

Then you have the category of insiders, and those have been going on for a long time. They are typically disgruntled insider employees who have too much system access, or retain access after they're terminated, and they decide to mess things up through logic bomb type scenarios, or just using their knowledge of system weaknesses and vulnerabilities to tamper with things afterward. Incidents like that happen all the time, but that doesn't normally make the news unless there's a huge civilian impact; but insiders are always a problem.

And then finally, you have commodity attacks. You get ransomware on an industrial network that's running Windows computers, and it's going to have some impact, especially if they're low-resource systems, systems that are used for interfaces for systems, HMIs, monitoring system behavior and safety, et cetera. Those can definitely have unforeseen impact.

For the more advanced adversaries, we always see a range of reconnaissance and basic foothold gathering in industrial systems, and then we see these novel attacks that occur more rarely against individual systems, especially in places that are used as test beds by advanced adversaries.

Investigating Cyber Criminals

Depending on your role, a cybercrime investigation may include exploration into the technical aspects of a crime (such as understanding how a hack occurred), or may include research and attribution of the criminals themselves.

A key piece of any cyber investigation will usually involve some level of understanding of the threat actors and how they operate. Especially during a breach, understanding the perpetrating group and their MO (*modus operandi*)

can provide crucial information, such as how they gained initial access, their network pivot points, which files they tend to examine, and more.

The examples and scenarios in this book are taken from my real-life experience investigating and interacting with The Dark Overlord (TDO) and Gnostic Players groups. Many of the examples throughout this book's chapters may mention the group as a collective, its individual members, or one of their many aliases.

The Beginning of the Hunt (for TDO)

One day, circa 2017, a friend told me that one of their affiliates was extorted by "The Dark Overlord," and asked if I knew anything about them. At that point in time, I remember the group being all over the news for trying to extort Netflix, and threatening to release advance copies of *Orange Is the New Black*, and hacking/selling data stolen from dozens of medical providers.

On that day almost two years ago, I threw myself into the darkweb of the cyber criminal underground. This was not a paid endeavor; it was more like that fun new hobby that you can't get enough of.

Tracking the group over the last few years has been one of the most exciting and rewarding experiences of my life. I have met so many interesting people, and have learned so much about technology and human behavior along the way.

In a way, I suppose I should be thanking them. If it weren't for the group's childish, illicit, and attention-whoring behavior, I would not have gone down this path, and would not be writing this book.

The Dark Overlord

In 2016, a hacking group known as The Dark Overlord (TDO) began terrorizing and extorting organizations. The group quickly became known throughout the media from the large number of hacks on medical providers. Some of their first publicized hacks included Midwest Orthopedic Pain & Spine Clinic in Farmington, MO; Midwest Imaging Center, LLC; and Van Ness Orthopedic and Sports Medicine.

As previously mentioned, the group gained additional headlines in 2017 for hacking Netflix and threatening to release advance copies of *Orange Is the New Black* if their ransom demands were not met.

Later that year the group moved from traditional "hacking" to more terror-based attacks. The group sent death threats to the parents of students in the Columbia Falls, Montana, school districts, causing the closure of more than 30 schools and forcing more than 15,000 students to stay home for an entire week.

The group would regularly post their demands and latest hacks over Twitter, with supporting information on Pastebin. The image in Figure 2.2 is the group's official avatar, a painting by Syrian artist Aula Al Ayoubi.



Figure 2.2

The following sections will provide summary information on the group's history, its motives, and information on each of its members.

List of Victims

The following organizations have publicly announced being attacked or extorted by The Dark Overlord. This is not an exhaustive list of all TDO's victims. A number of organizations have not publicized their involvement with the group.

NOTE *Being on this list does not imply a successful breach or ransom. It simply means there is a public record of the organization being hacked or terrorized by the group in some way. DataBreaches.net has maintained an excellent public record of the group's many hacks.*

- | | |
|--|--------------------------------|
| ■ A.M. Pinard et Fils, Inc. | ■ Athens Orthopedic Clinic |
| ■ ABC Studios / Steve Harvey | ■ Auburn Eyecare |
| ■ Adult Internal Medicine of N. Scottsdale | ■ Austin Manual Therapy |
| ■ Aesthetic Dentistry | ■ CB Tax Service |
| ■ All-American Entertainment | ■ Coliseum Pediatric Dentistry |
| ■ American Technical Services | ■ Columbia Falls, MT |
| | ■ Disney Studios |

- Dougherty Laser Vision
- DRI Title
- Family Support Center
- Feinstein & Roe
- Flathead Falls School District
- G.S. Polymers
- Gorilla Glue
- H-E Parts Morgan
- Hand Rehabilitation Specialists
- Hiscox (Hoax)
- Holland Eye Surgery and Laser Center
- Indigofera Jeans
- International Textiles & Apparel
- Johnston Community School District
- La Parfumerie Europe
- La Quinta Center for Cosmetic Dentistry
- Line 204
- Little Red Door Cancer Services of East Central Indiana
- Lloyd's of London
- London Bridge Plastic Surgery
- Marco Zenner
- Menlo Park Dental
- Mercy Healthcare
- Midwest Imaging Center
- Midwest Orthopedic Clinic
- Mineral Area Pain Center
- Netflix / Larson Studios
- OG Gastrocare
- PcWorks, L.L.C
- Peachtree Orthopedic Clinic
- Photo-Verdaine
- PilotFish Technology (PFT)
- Pre-Con Products
- Prosthetic & Orthotic Care
- Purity Bakery Bahamas
- Quest Records, LLC
- Royal Bank of Canada
- Saxon Partners
- School District 6
- Select Pain & Spine
- SMART Physical Therapy
- St. Francis Health System
- Tampa Bay Surgery Center
- UniOptics, L.L.C
- Van Ness Orthopedic and Sports Medicine
- WestPark Capital

A Brief Overview

It is believed that the group initially gained remote desktop protocol (RDP) access to their first medical clinic by purchasing access from Xdedic, a darkweb marketplace that sold inexpensive access to hacked computers around the world.

From there, it is believed that the group was able to gain access to further clinics and medical facilities by way of the HL7 medical software. In an interview with databreaches.net, a spokesperson for TDO made the following statement:

“... I used [HL7’s] code to find exploits in all their clients. ... Also, since I was in their system, I signed a backdoor into their client—because I had access to their certificate signing. It got pushed out in an update a few weeks ago.”

The victimized companies ranged in size and industry, and were often asked to pay excessive amounts of money in exchange for not having their confidential documents published on the Internet.

In addition to the hacking of medical facilities, the group also regularly targeted plastic surgery clinics, threatening to release pre-/post-op breast augmentation photos of famous Hollywood patients.

TDO would hold true to their word and release information if their demands were not met. Most recently, the group leaked nude photos of Frankie Essex when London Bridge Plastic Surgery refused to pay the group’s ransom demands.

Communication Style

Official communication from the group is standardized against “The Queen’s English.” The group members were strict to maintain this dialect throughout all their communications. Direct communication with TDO always contains a high level of grandiosity, especially when discussing business, or their “hacking skills.”

The characters maintain the persona of working within a large organization, and regularly refer to themselves as “we.”

TDO took great care in describing their business savvy, and would often describe the success of the “brand” they created.

When the group first started in 2016, communications via their Twitter account appeared to come from someone with broken English.

As the group matured, they developed a more formal persona with standardized formal English, perhaps to give the appearance they were from the UK.

Following their change of leadership in 2017 the overall tone of the group became increasingly hostile. Communications with victim organizations became significantly more aggressive, as did the group’s tone on Twitter.

This transition also marks the group’s tendencies toward less traditional hacking and more terror-based attacks, as witnessed by their behavior in sending death threats to the students of the Columbia Falls school districts.

Group Structure and Members

I believe The Dark Overlord group consisted of four core members and a small network of “contractors” used to carry out menial tasks. Evidence also suggests The Dark Overlord group is loosely and unofficially led by Cyper, the admin of the KickAss forum.

The group was formed circa 2015, where each of the members met on Hell, an old darkweb hacking forum. Arnie was the group's initial public leader, while Cr00k appeared to be the person in charge of selling the stolen data and overall marketing.

Each of the group's members played a role in hacking the various victims. In 2017, TDO publicly announced over Twitter that it was under new leadership. Following this transition, it is believed that leadership of the group was transferred to NSA(@rows.io).

The following is an excerpt of a private conversation with the original TDO (2016) via the group's Twitter account (@tdohack3r):

```
you know i am not alone?
i have team
we have a expert english speaker for ransom
i do hacks
others steal the data
i am good at exploit and attack
partner is good at english and business
another is good at stealing data ad running backup and server making
ransomware
```

Cyper

The group's unofficial leader goes by the name Cyper (no H) or CyPeRtRoN. He is the oldest member of the group (in age) with a very strong understanding of C++ and other coding languages. He is easily recognizable due to his broken English. To the best of my knowledge he lives in Austria and is in his 40s.

While he may not be a "leader" in the traditional sense, he was certainly a mentor to the other group members, and is present with the group members through this entire story.

Between 2002 and 2013, CyPeRtRoN appeared to spend most of his free time publicly defacing websites. A list of his 208 total website defacements can be seen using the zone-h defacement archive at <http://www.zone-h.org/archive/notifier=CyPeRtRoN>.

By his own account, Cyper's most reputable defacement was the 2003 alteration of the U.S. Navy's OWA site (Figure 2.3): <http://zonehmirrors.org/defaced/2003/03/04/owa.navseadn.navy.mil/Cy.jpg>.



Figure 2.3

Cyper is also the known admin of the very exclusive BlackBox forum (under the name Ghost), and the more recent KickAss forum (under the name NSA).

Cyper's other aliases include:

- Cypertron
- NSA
- Ghost
- 100k
- 100k2

NOTE Important to note—the group members often interchanged aliases to create confusion. In this case, NSA, leader of the KickAss forum, took over the alias of NSA(@rows.io), who has been directly associated with several TDO-related hacks, including the Louisiana DMV (which he openly admits in a chat. When referencing TDO's NSA, I will reference him as NSA(@rows.io), to denote the his jabber address. I will refer to the admin of KickAss forum as Cyper whenever possible.

Arnie

Evidence suggests that Arnie is Nathan Wyatt (aka CraftyCockney), a 30 year old resident of the United Kingdom.

Evidence also suggests Wyatt was involved with TDO at the onset of the group's formation in 2016, he was the original persona of The Dark Overlord (the one who would speak in broken English), and was also the group's original lead figure under the alias Arnie.

Arnie first gained media attention by announcing the original TDO healthcare hack and sale of the related medical data on the Hell Reloaded forum (Figure 2.4).



Figure 2.4

Arnie was listed as the primary seller of similar data on other forums, and would openly advertise that he was the person hacking into the healthcare clinics on various forum posts.

On September 24, 2016, Wyatt was arrested on suspicion of Computer Misuse Act offenses for attempting to broker the sale of pictures of Pippa Middleton that were hacked from her iPhone.

In December 2016, following a search of Wyatt's devices, the London Metropolitan Police Service found evidence of thousands of stolen documents from a UK law firm previously extorted by TDO.

Following his arrest, Wyatt agreed to be interviewed by Dissent Doe, reporter for DataBreaches.net.

During the interview,

Wyatt describes his relationship with The Dark Overlord, which included teaching thedarkoverlord fraud techniques, and being asked by TDO to make an extortion phone call to a U.S. victim.

Additional details to the exclusive interview were published by Vocativ.com (the following quotes both can be found at <https://www.vocativ.com/362147/pippa-middleton-hack-photos-arrest-uk>):

Crafty says that he is not a hacker and is acting solely as an agent for the individual(s) wanting to sell the material. "I'm not a hacker," he said, "Neither did i commit the hack. I'm of the community, but in all honesty I surprise myself I managed to get simple encryption working."

Wyatt also describes the group's technique to use the media to drive up the price of their extortion demands for the Pippa Middleton photos:

The UK press played into our hands. I had no intention of selling them anything. But they have printed their authenticity. That's half the difficulty done for me . . .

Indictment and Extradition

Wyatt has been extradited to the United States on several charges related to crimes associated with The Dark Overlord group.

The following excerpt of the charges against Wyatt was obtained from the official court indictment and extradition document (which Wyatt was kind enough to send me):

- Victim funds were transferred to a PayPal account with email tashiadsmith@tutanota.com
- Threatening extortion text messages were sent from 337-214-5137, which was registered using Wyatt's home IP address

- Wyatt registered a WhatsApp account with 337-214-5137 and uploaded his photo as the avatar
- The same number was used to log into the PayPal account that received the victim payments (tashiadsmith) and was set up as the home phone for that account
- An extortion email sent to a victim requested funds be split into four different UK bank accounts. The emails included bank account information in Wyatt's name and his girlfriend's name
- A UK number, 44 775-481-6126, was registered to Nathan Fyffe. That number was used to register Wyatt's personal Facebook account and the @TDOHACK3R Twitter account used by the group
- The same phone number was used to order pizza for home delivery to Wyatt's home address and was also used to register a VPN service that was used to log into the above

The sheer amount of evidence against Wyatt is hard to ignore. I have no doubt that Wyatt was involved with the other members of TDO during their first year of operation. I also think his association with the other members made him the perfect candidate to take the fall for the other group members.

NOTE As we will discuss in future chapters, a specific characteristic of the other group members is their ability to find a patsy, or someone on whom they can pin all of their crimes.

The indictment states that Wyatt willingly used bank accounts in his *and* his girlfriend's name, which were to be used to cash out money earned from the group's extortion schemes. This is the one charge that I personally find a bit hard to believe. According to the indictment, the account numbers were used in a TDO extortion email.

Given the MO of the other group members, it is very possible they were able to find his account numbers and intentionally include them in an email to a victim.

That being said, Wyatt *willingly* recorded and published an audio clip on YouTube of himself *rapping* a threatening voicemail/demand for payment to one of the group's extortion victims. During the call he even referenced himself as part of the "The Dark Overlords."

You can listen to the audio here: <https://youtu.be/DzApepLbA70>.
Arnie's other aliases include:

- CraftyCockney
- craftycockn3y

- Gingervitis
- JasonVoorhees
- l00t5
- Mari0

Cr00k (Ping)

Cr00k's role in TDO was to act as their primary salesman/data broker. Cr00k has held a number of different aliases over the years. He originally gained notoriety in 2015 when he was outed as "Ping," the owner and admin of Hell Forum, an underground darkweb hacker forum.

In addition to all of these accomplishments, he is also an incredibly skilled hacker, with a very deep understanding of different technologies.

Cr00k likes to create confusion and deception by Hijacking the aliases of well-known hackers, and will often use the press to gain attention for his hacks and the merchandise he is involved with trafficking.

He is also a master at using the media for manipulating events and throwing law enforcement off his trail.

And he is only 18 years old.

In my opinion, he has a gift for long-term planning and strategy, by creating fake scenarios, publishing fake doxes, and manipulating conversations to create stories designed to send investigators down endless rabbit holes of incorrect information.

Cr00k's other aliases include:

- C86x
- Dio_the_plug
- F3ttywap
- Frosty
- Jinn
- Lava
- Nakk3r
- NSFW
- Malum
- Ping
- Photon

- Prometheus
- Overfl0w
- Rejoice
- ROR[RG]
- Ryder
- Russian

NSA (Peace of Mind)

NSA, nsa@rows.io (not to be confused with Cyper, who is also known as NSA), was only active under that name for a few months.

NSA became known for advertising the sale of hacked records from the Louisiana Department of Motor Vehicles on TheRealDeal market. His origins can be traced back to 2015, when he was an admin of the original Hell forum as “Revolt,” and TheRealDeal market as “Peace of Mind.”

Evidence suggests that NSA is the true persona behind The Dark Overlord and, as referred to by Nathan Wyatt, the “kid” that took over as the group’s leader.

I believe his greatest strength is his ability to deceive and create confusion, which he does under the guise of multiple aliases. I have read numerous confrontational conversations that he has had with himself under different aliases just to create confusion and misinformation.

Any reporters or investigators that had a conversation with TDO after 2017 were most likely speaking with him.

NOTE Ping and Revolt, also known as Cr00k and NSA, live about 5 miles from each other in Calgary, Canada. They grew up together and were regularly involved in hacks together dating back to the Hell forum.

TDO’s New Leader

In 2017, TDO announced a leadership change over Twitter, with a tweet titled “New year, new us.” The tweet included a Pastebin link of pastebin.com/kekU-JRU7 that has since been removed (and unfortunately not in any archives). This change marks the end of Arnie as TDO’s leader, and the beginning of a new regime under nsa@rows.io.

Pre-2017, TDO’s attacks were primarily focused on gaining access through RDP servers and hacked medical software. Following the leadership change to NSA, the attacks became hostile and much more aggressive.

TDO's language also became much more aggressive and abusive, and their style of attacks moved from typical technology-based hacks to outright terror. Later that year, the group was placed on a national terror list after they forced the closure of several school districts by sending death threats to the parents of students.

Figure 2.5 is a screenshot of text messages sent to one student's parents.

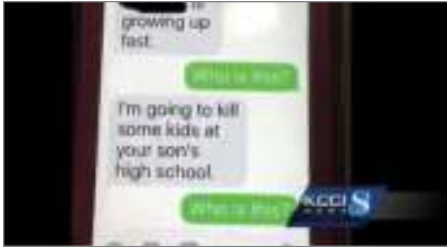


Figure 2.5

Another example of the group's cruel nature involved the extortion of "Little Red Door," an Indiana-based nonprofit cancer clinic. NPR.COM reported that TDO's communication with the clinic came via email, with a subject line of "Cancer Sucks, But We Suck More!"

When Little Red Door did not pay the almost \$50,000 ransom, TDO took to Twitter and published grief letters sent to families of clients that passed away.

Attitude and Hostility

In addition to his overly hostile tone, NSA's history on hacker forums shows a tendency toward gay-bashing, while his personal social media pages show a number of pro-homosexual images and references.

I have had the pleasure of speaking to him many times via his personal Facebook, Twitter, and Jabber accounts. The following is a small excerpt that shows his general attitude (note the constant use of the word d**k):

```

XXX    I'm sorry man but you sound like you're either LE or a f***ing
        retard, probably the latter.
VT     why am i a retard?
XXX    you and your friends NightCat/jasonvoorhees/hafez asad and other
        d**kheads can go climb a wall of d**ks
VT     NightCat? Jasonvoorhees?
XXX    s u c k a d * * k
  
```

Some of NSA's other aliases include:

- BTC
- Columbine

- L3tm3
- NSFW
- Obbylord
- Obfuscation
- Peace of Mind
- Revolt
- Stradinatras
- WhitePacket
- Vladimir

The Dark Overlord

Whenever communicating as the TDO figure, there appeared to be language guidelines and a strict formal tone in order to give the appearance of a larger group structure with internal processes and procedures.

Something I have always found comical is the effort TDO will place in trying to present themselves as a legitimate business. In numerous conversations with the group members, they always discuss contracts, invoices, or other formal business documents needed to employ their services.

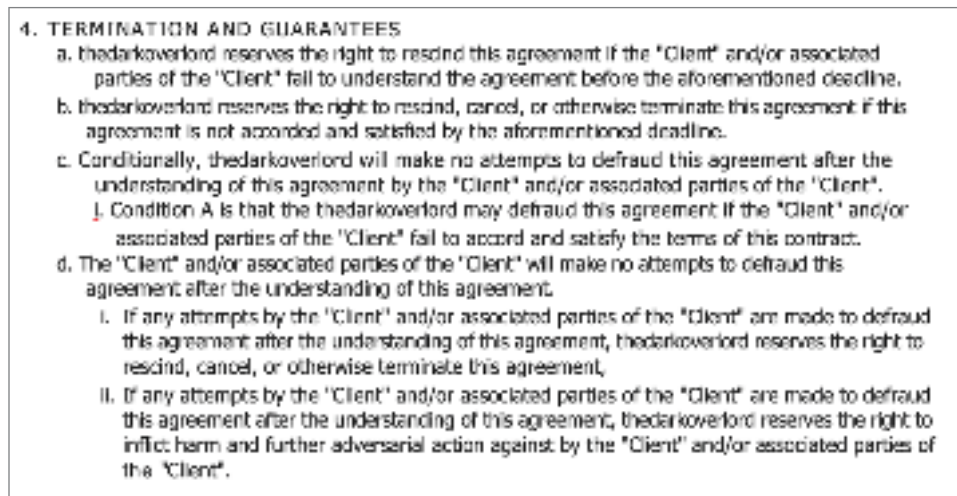


Figure 2.6

Figure 2.6 shows a section of an alleged contract between The Dark Overlord and a victim, published by Joseph Cox on Motherboard.

Despite appearances, the group's businesslike tone quickly degraded if a victim did not pay their ransom. Whenever TDO's ludicrous demands were not met, the group would quickly become agitated and throw a public tantrum



Figure 2.7

over Twitter, no different than the behavior of a child not getting what they want (Figure 2.7).

The following is an excerpt from a three-hour conversation I had with TDO over Jabber chat in October 2018. TDO knew he was speaking to me, Vinny Troia.

Additional conversations and excerpts will be analyzed throughout this book as they relate to the material in different chapters. These excerpts provide an excellent look into TDO's aggressive, self-aggrandizing, and profane personality:

12:27 AM	TDO	Dumb pr**k.
12:27 AM	TDO	You're pleased with connecting one of the rare few of us.
12:31 AM	TDO	You're fortunate, tonight.
12:32 AM	TDO	I'm sitting here moving terabytes of data into a new server, and I'm bored, so you've been blessed with communicating with me.
12:52 AM	TDO	It's quite nice to finally speak with you, Troia.
12:53 AM	VT	You too.
12:53 AM	TDO	Listen, the lads are cultured carefully to understand the linguistic systems that we deploy, so you'll forgive their brevity.
12:54 AM	TDO	Again, this is a rare opportunity, don't spoil it.
12:59 AM	TDO	Listen mate, forget Krebs... Do you realise that your nation-state has investigated us without success?
1:00 AM	TDO	It's been publicly reported. Check the Billings Gazette in Montana. It was leaked that the CIA and NSA have investigated us.
1:01 AM	VT	congratulations on evading their efforts
1:01 AM	TDO	Your NSA and GCHQ is not as good as they want people to believe. Passive surveillance is only so successful.
1:02 AM	VT	i dont doubt that

1:02 AM TDO Do you realise that we closed down six different school districts in Montana for 5 business days?

1:02 AM TDO We closed out 36.000 students from school.

1:03 AM TDO Now we've all heard about 'bombthreats' closing schools for a day, but what does someone have to do to close schools for 5 days?

1:03 AM TDO Have you ever thought about this?

1:03 AM VT honesly, no

1:03 AM TDO You're an idiot then. What other organisations have closed an entire region for that long?

1:04 AM TDO Think about it. What did we need to do to achieve this goal?

1:04 AM TDO 5 days. Not 1, but 5.

1:04 AM TDO Ponder it. We actually had your FBI physically chasing us,

1:04 AM TDO Chasing ghost.s

1:04 AM VT I didnt realize the schools were closed for an entire week. You are right, that is pretty significant

1:05 AM TDO You should read into it. It's all OSINT.

1:05 AM VT refresh me on the story, you guys were calling the school, correct?

1:05 AM TDO Calling? Far beyond that.

1:05 AM TDO We planted physical devices using unassuming third-parties.

1:06 AM VT i have no idea what that means

1:06 AM TDO F***ing moron.

1:06 AM TDO Think.

1:06 AM TDO Use that f***ing bit of grey matter between your eyes.

1:06 AM VT what is an unassuming third party

1:06 AM TDO This is why your NSA and CIA investigated us, and conducted raids in London.

1:07 AM TDO Mind you, unsuccessful raids.

1:07 AM VT I did not know any place was raided.

1:07 AM TDO READ THE BILLINGS GAZETTE, the great piece of OSINT on TDO EVER.

1:07 AM TDO F***ing moron, Troia.

1:07 AM VT Sorry, the Billings Gazette was not exactly high on my PR radar

1:08 AM TDO F***, I'm sitting here on this evening speaking with the biggest moron on this f***ing rock.

Getting TDO worked up was entertaining as it was useful. The ability to trigger his rash emotional outbursts would come in handy later on when I would speak to other aliases. As long as you have the stomach for it, being able to trigger the same outbursts simply asking the right questions is an incredibly efficient way to gain a better understanding of who you are speaking with.

Summary

This chapter provided an overview of some of the different paths available as an investigator, and some of the scenarios that I have personally been involved in on my quest to help inform organizations of a breach.

This chapter also provided background information on The Dark Overlord and its core group members, which will be discussed in greater detail in later chapters.

Network Exploration

In This Part

Chapter 3: Manual Network Exploration

Chapter 4: Looking for Network Activity (Advanced NMAP Techniques)

Chapter 5: Automated Tools for Network Discovery

As the title suggests, this part of the book discusses many of the intricacies of network investigation and exploration. Investigators and black box penetration testers will often have nothing more to go on than a single IP address, so that's where our investigative journey begins. These next three chapters will look at processes and tools designed to thoroughly explore the network space and help to gather as much information as possible.

Manual Network Exploration

This chapter is designed to give you an overview of performing general network reconnaissance on an organization, as well as scanning and identifying network hosts. The focus of this chapter is more on the process than the specific tools being used because when it comes to OSINT, or probably any facet of information security, there will always be newer, better, and shinier tools to use.

There are hundreds of different tools and services you can use to gather information on organizations. New tools come out almost daily; it's impossible to keep up. I don't want that to be the focus. I'm sure I missed some tools, and I'm sure some will be outdated by the time you read this. My goal is to show you some tools that I think are useful *now*, but I completely understand that there may be better ones in the future.

Regardless, the tools should not be the focus. The point is *how* the information is being collected, and more importantly, how the process of discovering new information should *never* be limited to just one tool.

Always try new things because, as I will demonstrate, one tool will never give you a complete answer.

Chapter Targets: Pepsi.com and Cyper.org

To illustrate the differences between the different tools and services, I will run all tests against two different servers, pepsi.com and cyper.org. I am showing two very different sites to show the differences in the software output when working with two very different targets.

Why pepsi.com and cyper.org? [Pepsi.com](http://pepsi.com) because I am drinking one as I write this. [Cyper.org](http://cyper.org) because of the name. As far as I know, cyper.org is not affiliated with the threat actor Cyper (who I believe is affiliated with The-DarkOverlord hacking group). During the course of any investigation, new avenues of research like this will come up, and having to research uncommon domain names (like cyper.org) will be necessary.

Pepsi is obviously a very well-known brand and popular website, so the results between the two websites will be very different.

NOTE Before we start, I would like to point out that IP attribution is probably one of the most difficult tasks faced by investigators. To give you an example, I have discovered numerous data leaks, including Exactis, Apollo.io, and Verifications.io. In each of those cases, all I had to work with was the IP address of the leaky database server. Database servers don't usually refer back to a primary domain, so it's up to the investigator to find true attribution to the data. I will provide tips along the way with references to actual situations I have dealt with. Hopefully my experiences will help some of you along the way.

Asset Discovery

Getting started with network exploration is usually pretty straightforward. If you are performing a black box test, then you know the client's name and their domain. To get their IP address, you can perform a Dig lookup. Dig is included on Linux/Mac by default. Running the following command will give you the IP for apple.com:

```
root@INTEL:~: dig apple.com

; <<>> Dig 9.10.3-P4-Ubuntu <<>> apple.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 7666
;; flags: qr rd ra; QUERY: 1, ANSWER: 3, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;apple.com.                IN      A
```

```
;; ANSWER SECTION:
apple.com.      1100  IN      A       17.172.224.47
apple.com.      1100  IN      A       17.142.160.59
apple.com.      1100  IN      A       17.178.96.59

;; Query time: 1 msec
;; SERVER: 8.8.8.8#53(8.8.8.8)
;; WHEN: Mon Dec 10 09:02:24 UTC 2018
;; MSG SIZE rcvd: 86
```

ARIN Search

Having one IP address is great, but we would like to know more about the target's IP space. The American Registry of Internet Numbers (ARIN) manages the distribution of IPv4 and IPv6 addresses. Many large organizations will register entire net ranges, so their IP space will be easily searchable. ARIN provides a free search to quickly identify these IP ranges.

The WHOIS protocol (which will be discussed in Chapter 9) is used to query databases containing all sorts of publicly available information on Internet resources, including domain names and IP addresses. The WHOIS protocol is used to query a wide network of WHOIS servers for any information on the domains behind the billions of websites around the world (collectively known as WHOIS data).

WhoisRWS search looks at data specifically within the ARIN's registration data. This can often be a good first step when performing a black box assessment and all you have to go on is a company name. Figure 3.1 shows an example ARIN search window.



Figure 3.1

Entering “Google” in the WhoisRWS search gives us a few different network ranges we can add to our searches, as shown in Figure 3.2.

NOTE This is a great first stop for me because sometimes you will get lucky and find that an IP address (or even an entire IP block) is owned by a particular owner. When you're looking for server or database attribution, it's rare that this happens, but I've been lucky a few times. There was one data leak in particular I want to mention. A Chinese telecom company had over 3 TB of data leaking, including customer names, email addresses, mobile numbers, and debit card numbers. Luckily, they owned their entire netblock, so attribution was fairly simple. It doesn't always happen this way, but it's worth trying first.

You searched for: google

Networks	
GOOGLE (NET-2620-E7-4000-1)	2620:E7:4000:: - 2620:E7:4000:FFFF:FFFF:FFFF:FFFF:FFFF
GOOGLE (NET-2620-E7-C000-1)	2620:E7:C000:: - 2620:E7:C000:FFFF:FFFF:FFFF:FFFF:FFFF
GOOGLE (NET-108-170-192-0-1)	108.170.192.0 - 108.170.255.255
GOOGLE (NET-108-177-0-0-1)	108.177.0.0 - 108.177.127.255
GOOGLE (NET-142-250-0-0-1)	142.250.0.0 - 142.251.255.255
GOOGLE (NET-172-217-0-0-1)	172.217.0.0 - 172.217.255.255
GOOGLE (NET-172-253-0-0-1)	172.253.0.0 - 172.253.255.255
GOOGLE (NET-173-194-0-0-1)	173.194.0.0 - 173.194.255.255
GOOGLE (NET-192-178-0-0-1)	192.178.0.0 - 192.179.255.255
GOOGLE (NET-199-87-241-32-1)	199.87.241.32 - 199.87.241.63
GOOGLE (NET-199-88-130-0-1)	199.88.130.0 - 199.88.130.255
GOOGLE (NET-199-89-220-0-1)	199.89.220.0 - 199.89.220.255
GOOGLE (NET-207-223-160-0-1)	207.223.160.0 - 207.223.175.255
GOOGLE (NET-209-170-110-128-1)	209.170.110.128 - 209.170.110.191
GOOGLE (NET-209-170-110-192-1)	209.170.110.192 - 209.170.110.255
GOOGLE (NET-209-170-119-128-1)	209.170.119.128 - 209.170.119.191

Figure 3.2

Search Engine Dorks

Finding active subdomains is a fairly easy job to do with Google dorks (advanced search strings).

NOTE A “dork” is a search string that uses advanced operators to generate new results. Search dorks will be covered in much greater detail in Chapter 8. For now, this section covers some really simple dorks that can be used to find information on a specific website address.

To search for results within a specific website, just use the following command:

Site:pepsi.com

Now we can further enhance the search result to remove subdomains that we don’t want (e.g., www), by using “-inurl” (minus inurl). Inurl will search for a string within a URL, and we want to do the opposite—remove anything in the URL that matches our string:

Site:pepsi.com -inurl:www



Figure 3.3

There are 609 page results, and the first page alone (Figure 3.3) looks like it has targets worth investigating.

DNSDumpster

People are busy. The busier they are, the more forgetful they tend to become about cleaning things up. When looking into DNS records or even simply looking for more information about a target, I am always able to find information on subdomains that should have been decommissioned. It is very likely that developers will create “test” subdomains for development projects not yet pushed to production. Developers aren’t always great at cleaning up their code and often employ shortcuts, especially when working on their own development sites. These subdomains and shortcuts can often provide a gold mine of information.

You can search for these by using DNSDumpster, a free domain search tool that can quickly discover related hosts by IP address.

Many times a subdomain is left pointing to an old IP address. I've had situations where a company subdomain was pointing to an IP address that was not even owned by the company. The IP was to a random public FTP site, and unbeknownst to the bank, `blah.bank.com` was pointing to a subdomain that was holding pirated software. They were very thankful for the find.

DNSDumpster couldn't be easier to use (see Figure 3.4). Just put in a domain or IP you are looking for and it will display a really nice graphical output for you.

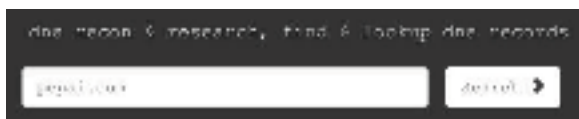


Figure 3.4

Figure 3.5 shows an example output using `pepsi.com` as our target. DNSDumpster gives us 19 total host records. The following is only a small subset of the results, but from here we can already see four different potential target IPs, all with interesting-looking names (`points`, `account`, `shop`, and `secure`).

Host	IP Address	Additional Info
www.pepsi.com	74.125.142.107	201403 - Google, Failed Status
account.pepsi.com	104.145.145.113	201403 - Google, Failed Status
shop.pepsi.com	191.13.107.18	201403 - Google, Failed Status
secure.pepsi.com	16.210.129.69	201403 - Google, Failed Status

Figure 3.5

There is also a free download to XLS option that gives you a great breakdown that you can easily import into another program (see Figure 3.6).

Figure 3.7 shows a nice graphing option that will create a network map of the discovered devices. This is a great free tool to use in your penetration testing reports for a little extra “wow” factor.

When we run the same search against our second domain, `cyper.org`, we get 1 resulting host record:

```
www.cyper.org - 162.210.102.59
```

[illegible]

Figure 3.6

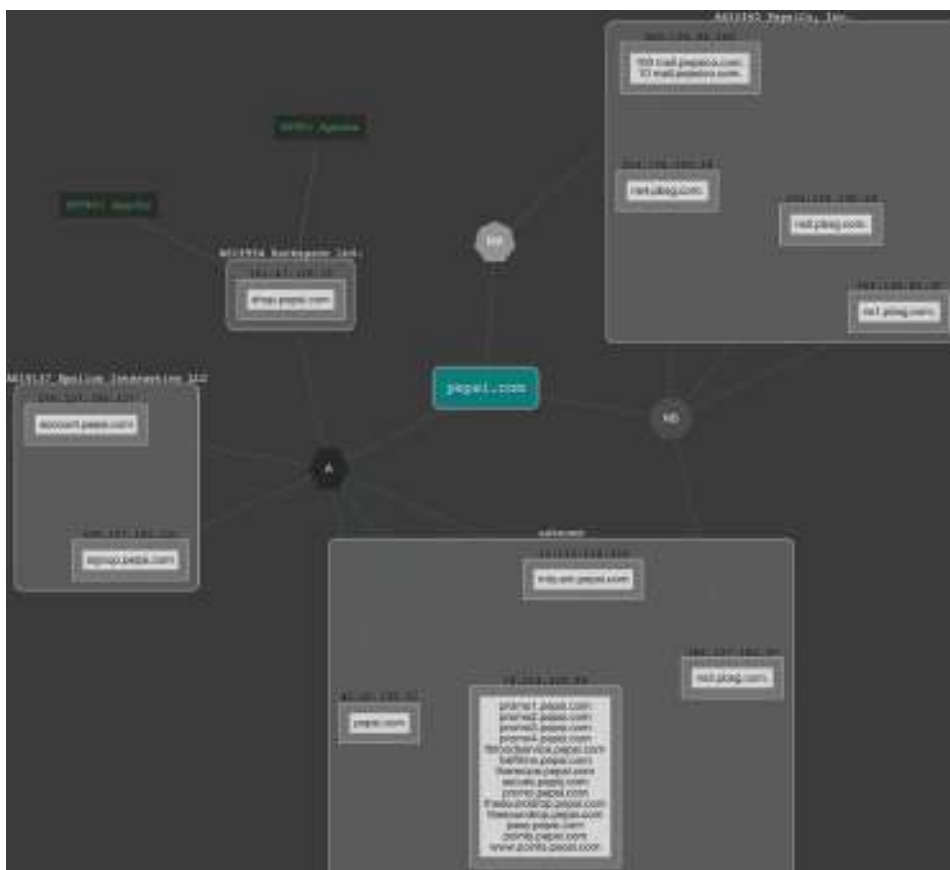


Figure 3.7

Hacker Target

Hacker Target (<https://hackertarget.com/reverse-dns-lookup>) makes searching for subdomains and reverse DNS lookups really easy using their API. You can just query a domain or IP with a single URL like this:

```
https://api.hackertarget.com/reversedns/?q=url.com
```

A quick search for `pepsi.com` on `hackertarget.com` reveals 31 different subdomains:

```
https://api.hackertarget.com/reversedns/?q=pepsi.com
```

```
sadad scl-pepsi.com,124.29.219.220 58.27.215.220 124.29.219.221
moscntx3.russia.intl.pepsi.com,195.239.40.163
www.wpbpepsi.com,209.142.183.140
mail.hbcpepsi.com,210.56.13.100
```

```
mbcpepsi.com,156.46.83.96
router.mbcpepsi.com,69.29.19.133
mx.mbcpepsi.com,69.29.19.134
mail.nbcpepsi.com,116.58.32.20
sticladepepsi.com,216.247.113.7
lakesidepepsi.com,71.13.1.74
iwanttoservepepsi.com,165.197.98.42 204.136.80.125
host170.mbgpepsi.com,205.244.118.170
host171.mbgpepsi.com,205.244.118.171
mbg1.mbgpepsi.com,205.244.118.161
host162.mbgpepsi.com,205.244.118.162
mail2.mbgpepsi.com,173.219.102.86
host163.mbgpepsi.com,205.244.118.163
host173.mbgpepsi.com,205.244.118.173
host164.mbgpepsi.com,205.244.118.164
host165.mbgpepsi.com,205.244.118.165
host166.mbgpepsi.com,205.244.118.166
host167.mbgpepsi.com,205.244.118.167
host168.mbgpepsi.com,205.244.118.168
host169.mbgpepsi.com,205.244.118.169
mail.mbgpepsi.com,206.107.106.67
gjpex01.gjpepsi.com,173.226.224.197
cin_prd.gjpepsi.com,38.155.184.10
autodiscover.exchangedelegation.gjpepsi.com,173.226.240.194
domino.forumpepsi.com,207.245.55.250 207.245.55.251
mail.bremertonpepsi.com,70.90.189.233
viv.vivehoypepsi.com,174.120.246.142
```

A second search for `cyper.org` reveals five subdomains:

```
https://api.hackertarget.com/reversedns/?q=cyper.org
```

```
office.cyper.org,65.100.252.81
zone.cyper.org,65.100.252.84
seek.cyper.org,65.100.252.82
storm.cyper.org,65.100.252.83
www.cyper.org,65.100.252.85
```

Shodan

Shodan is a search engine for Internet devices. Shodan does a pretty amazing job with mapping devices across the Internet, so all you have to do is ask it the right question.

Shodan literally queries every IP address and its ports and collects information about the server's responses. What makes Shodan so great is that you can use it to drill down and look for specific targets of technology. For example, are

you interested to know which wind turbines have accessible IP addresses? Ask Shodan. Want to know which Elasticsearch or MongoDB databases can be accessed without credentials? Want to know which camera or home routers are still using their default password? You guessed it. Shodan has those answers.

NOTE Shodan only offers a limited number of free queries, so if you want to be able to search more and do cool things like use its API to download lists of data in bulk, you need to pay for access.

Shodan's search is very well designed and offers the ability to use advanced search operators (dorks) to quickly search for specific targets. Here are the basic filters you can use with Shodan:

- **city:** Find devices in a particular city
- **country:** Find devices in a particular country
- **geo:** Use geo coordinates
- **hostname:** Look for devices that match a particular hostname
- **net:** Search based on an IP or CIDR
- **OS:** Search based on operating system
- **port:** Look for specific ports
- **before/after:** Find results within a time frame
- **Org:** Search for a specific organization name

We can search for a specific organization like this:

```
Org: Company Name
```

To search for Pepsi as an organization, we can use the following search:

```
Org: Pepsi
```

If using the org search parameter is too broad, we can try to narrow down our searches by looking for any results with specific hostnames (see Figure 3.8):

```
Hostname:pepsi.com
```

This gives us 28 results associated with `pepsi.com` and is a good start.

If you can't search for a particular hostname in Shodan, or if you just want to be thorough, you can try searching for a domain's IP.

In the case of `cyper.org`, a hostname search returns 0 results. We can quickly find the IP associated with `cyper.org` using Dig:

```
root@OSINT: dig cyper.org
```

```
<<>> DiG 9.10.6 <<>> cyper.org
```

```

global options: +cmd
;; Got answer:
;; ->>HEADER<- opcode: QUERY, status: NOERROR, id: 12771
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
; cyper.org          IN          A

;; ANSWER SECTION:
cyper.org.  3155      IN          A          162.210.102.59

```



Figure 3.8

Now let's check Shodan for any hits based on that IP (also referred to as a reverse DNS lookup).

There are no subdomains listed in this output; instead we are provided with a significant amount of metadata from this IP, including other associated domains.

We will discuss reverse DNS (RDNS) lookups in greater detail in Part II of this book (website information gathering)—specifically when looking at domain names. As the name implies, a reverse DNS lookup takes an IP address and returns all domains associated with that IP.

Censys (Subdomain Finder)

Censys.io (www.censys.io) is a service very similar to Shodan in that they both provide visibility of devices connected to the Internet. Like Shodan, Censys is a tool that helps organizations understand what servers and devices they have exposed on the Internet. Censys.io can be used to search for assets on a particular network and even discover unknown hosts that may have been created by employees and forgotten about. The discovery of open infrastructure is a very important piece of an external penetration test because it identifies open ports or sources of information spillage.

In my opinion, the main difference between Shodan and Censys.io is that Censys provides historical data. This data comes at a price as Censys is considerably more expensive than Shodan. That being said, as we continue to discuss potential investigation avenues, it will become apparent that having access to historical data can often be the cornerstone of an entire investigation.

Censys Subdomain Finder

Censys Subdomain finder (<https://github.com/christophetd/censys-subdomain-finder>) is an easy-to-use a command-line tool that uses the Censys.io API to return a list of available subdomains. Since my results will often vary between Shodan and Censys (or between any other subdomain research tool), this is a great way to quickly query Censys to check for additional hits.

For technically minded readers, the script queries Censys.io to get a list of available subdomains using the certificate transparency (CT) logs. Certificate transparency logs are amazing and can be used for a number of different types of attribution. We will cover CT in much greater detail in Chapter 10.

Using our two sample cases, first, a search for `pepsi.com` reveals 42 unique subdomains:

```
root@OSINT: python censys_subdomain_finder.py pepsi.com
```

```
[*] Searching Censys for subdomains of pepsi.com
[*] Found 42 unique subdomains of pepsi.com in ~1.4 seconds
```

Next, let's search for `Cyper.org`:

```
root@OSINT: python censys_subdomain_finder.py cyper.org
```

```
[*] Searching Censys for subdomains of cyper.org
[-] Did not find any subdomain
```

We can already see that the results between the two services are very different. This is mainly due to the size of the two organizations, but is a reminder that different tools will yield different results.

Fierce

Fierce is a DNS enumeration tool that helps locate noncontiguous IP space and hostnames against specified domains.

It's really meant as a precursor to NMAP since that already requires you to know what IP space you are looking for.

A tool doesn't need to have 100 options to be great. Fierce is meant for DNS enumeration and it simply does what it says it will do. The options include:

- **-dns-server:** Specifies DNS servers to use
- **-domain:** Domain to test
- **-subdomain-file:** Use subdomains specified in this file (one per line)
- **-range:** Scan an internal IP range, use cidr notation
- **-delay:** Time to wait between lookups

To keep things simple, let's run Fierce against our two example domains. We can also output the results to a text file using the Linux output command.

```
fierce --domain pepsi.com > output.txt
```

The result of the fierce scan shows six unique subdomain/host entries:

```
159.127.187.12    e.pepsi.com
18.214.229.99     promo.pepsi.com
18.214.229.99     secure.pepsi.com
161.47.125.72     shop.pepsi.com
159.127.185.121   signup.pepsi.com
209.143.254.67    stage.pepsi.com
```

```
Subnets found (may want to probe here using nmap or unicornscan):
```

```
159.127.185.0-255 : 1 hostnames found.
159.127.187.0-255 : 1 hostnames found.
161.47.125.0-255  : 1 hostnames found.
18.214.229.0-255  : 2 hostnames found.
209.143.254.0-255 : 1 hostnames found.
```

```
Found 6 entries.
```

Now let's try with our other domain:

```
root@OSINT: fierce --dns cyper.org --threads 5 --file dns.txt
```

The output of this scan is:

```
Checking for wildcard DNS...
Nope. Good.
Now performing 1917 test(s)...
198.23.53.116      imap.cyper.org
198.23.53.116      pop3.cyper.org
198.23.53.116      smtp.cyper.org
162.210.102.59     www.cyper.org

Found 4 entries.
```

Not a bad start. We are already four subdomains farther than with the previous scan. Let's keep going.

Sublist3r

Sublist3r is probably my favorite of all the hundreds of subdomain brute-forcing tools. Running the script with just a domain name will check for subdomains across a number of different engines, including Yahoo, Bing, Netcraft, DNSDumpster, PassiveDNS, and more.

I have used Sublist3r for years, and I know many of my hacker buddies also swear by this. I also try to keep up with the trends of the bug bounty hunters, and Sublist3r still seems to be a favorite. It's an incredibly simple tool to use, and the breadth of search results is impressive.

Sublist3r includes the following options.

- -d: Domain to search
- -b: Enable bruteforce module
- -p: List of ports to scan
- -v: Verbose
- -t: Number of concurrent threads
- -e: Specify which search engines to use
- -o: Output to file

Sublist3r does not require a lot of configuration or tweaking. You can just run it with the standard -d option to search a domain and let it do its magic:

```
root@OSINT: python sublist3r.py -d pepsi.com

[-] Enumerating subdomains now for pepsi.com
[-] Searching now in Baidu..
[-] Searching now in Yahoo..
```

```
[~] Searching now in Google..  
[~] Searching now in Bing..  
[~] Searching now in Ask..  
[~] Searching now in Netcraft..  
[~] Searching now in DNSdumpster..  
[~] Searching now in Virustotal..  
[~] Searching now in ThreatCrowd..  
[~] Searching now in SSL Certificates..  
[~] Searching now in PassiveDNS..  
[~] Total Unique Subdomains Found: 56
```

This initial query resulted in several hundred subdomains being returned from `pepsi.com`. Let's try the same command with our second domain:

```
root@OSINT: python sublist3r.py -d cyper.org
```

```
www.cyper.org  
office.cyper.org  
seek.cyper.org  
storm.cyper.org  
zone.cyper.org
```

```
[~] Total Unique Subdomains Found: 5
```

Now we've seen a handful of GUI and command-line tools that we can use to get lists of subdomains using more conventional search methods. Let's take this process one further step and try bruteforcing.

Enumall

Enumall is a script created by Jason Haddix to perform exhaustive subdomain discovery with a single command. Enumall is designed to do everything you've seen in this chapter with a single command.

Enumall uses the `recon-ng` engine to:

- Enumerate subdomains
- Perform Google, Bing, Yahoo, and Netcraft scraping
- Bruteforce to find subdomains
- Search Shodan
- Resolve all active addresses to IP addresses
- Output everything to a nice and neat CSV

The script uses a combination of wordlists to create different subdomain combinations to try during the bruteforce attempt. Before running the script, you should be aware of these specific parameters:

- `-a`: Use `alt-dns` to create a permutation of words based on a seed list
- `-p`: The seed list that we want to feed to `alt-dns`
- `-w`: The custom wordlist to use with `recon-ng` while bruteforcing

NOTE Using the `alt-dns` options might be overkill depending on the level of testing you want to perform. This will create millions of possible permutations to test and will often take a very long time to complete (12+ hours). Using the standard bruteforce dictionary with `-w` will probably be enough in most cases. If not, you can also try different dictionaries.

Let's first run the script against `pepsi.com`:

```
root@OSINT: python enumall.py pepsi.com -a -p ../altdns/words.txt -w \
/SecLists/sortedcombi-dnsrecon-fierce-reconng.txt
```

```
SUMMARY
```

```
-----
```

```
[*] 78 total (51 new) hosts found.
```

```
[*] 167 records added to '/opt/recon/domain/pepsi.com.csv'.
```

```
root@OSINT: python enumall.py cyper.org -a -p ../altdns/words.txt -w \
/SecLists/sortedcombi-dnsrecon-fierce-reconng.txt
```

```
[*] 5 total (5 new) hosts found.
```

```
[*] 6 records added to '/opt/recon/domain/cyper.org.csv'.
```

Results

Table 1.1 shows the results from the different tools we used.

Table 1.1: Subdomain search results

TOOL USED	PEPSI.COM	CYPER.ORG
DNSDumpster	19	1
Hacker Target	31	5
Shodan	28	0

TOOL USED	PEPSI.COM	CYPER.ORG
Censys	42	0
Fierce	6	4
Sublist3r	56	5
Enumall/Recon-ng	78	5

If you take anything away from this chapter it should be that not every tool is perfect, and each tool you use will often give you different (or varying) results. You should always use multiple tools in your arsenal. Scripts like Enumall are great because it will use multiple tools to give you the most possible results.

Phishing Domains and Typosquatting

Typosquatting is a technique of registering fake domains names that look similar to the original/legitimate domain name. It is really just a fancy way of saying it is a misspelled URL. Attackers will take advantage of the fact that people make spelling errors when they type, and will register fake domain names with the intent of harvesting credentials, distributing malware, or doing something malicious. Attackers will often use misspelled domains in phishing emails as a quick way to get someone to click a link.

NOTE Homoglyphs are also very popular now in phishing attacks. *Homoglyphs* are characters (or glyphs) that appear very similar. These often include characters from other alphabets that look very similar to the original character. For example, looking at `pepsi.com` and `pepsî.com`, can you tell the difference right away? The “i” is different.

Searching for phishing/typosquatted domains is typically useful for one of two reasons. Either 1) you want to include a list of potential phishing domains for a customer on an external pen-test report (which you should always do); or 2) you have been hired to socially engineer the customer and need to find a clever way to trick users into giving you their credentials.

The best tool I’ve found for sniffing out typosquatting on domains is DNSTwist.

Using DNSTwist is extremely straightforward and to the point. Just put in your desired target domain and it will generate a list of potential phishing domains and automatically check to see if they are registered and/or active.

You can also specify the following switches to enhance your results:

- -d: Use your own dictionary file
- -f: Output to file format (CSV, JSON)

- -r: Show only registered domains
- -w: Perform a WHOIS lookup on domains
- -b: Grab banners from hosts
- -a: Show all DNS records

Let's run it to only show registered domains and see the results:

```
root@OSINT: python dnstwist.py -r pepsi.com
```

Right off the top with minimal effort, we get 81 hits (the following results have been truncated):

```
Processing 936 domain variants ..... 81 hits
```

Addition	pepsia.com	185.18.82.122 NS:ns.dynamixhost.com MX:mx2.dynamixhost.com
Addition	pepsib.com	-
Addition	pepsic.com	78.41.204.29 NS:ns1.torresdns.com
Addition	pepsid.com	104.31.76.91 2606:4700:30::681f:4c5b NS:igor.ns.cloudflare.com
Addition	pepsie.com	199.59.242.151 NS:ns1.bodis.com MX:mx76.m2bp.com
Addition	pepsif.com	-
Addition	pepsig.com	184.168.221.50 NS:ns53.domaincontrol.com MX:mailstore1.secureserver.net
Addition	pepsih.com	-
Addition	pepsii.com	66.63.171.125 NS:ns1.ehostinginc.com MX:mail.pepsii.com
Addition	pepsij.com	-
Addition	pepsik.com	94.152.37.52 NS:ns1.kei.pl MX:pepsik.com
Addition	pepsil.com	NS:juming.dnsdun.com
Addition	pepsim.com	184.168.221.56 NS:ns65.domaincontrol.com MX:mailstore1.secureserver.net
Addition	pepsis.com	69.172.201.153 NS:ns1.uniregistrymarket.link
Addition	pepsit.com	162.255.119.194 NS:dns1.registrar-servers.com
Addition	pepsix.com	104.217.67.249 NS:v1.dns234.net
Addition	pepsiy.com	162.255.119.196 NS:dns1.registrar-servers.com
Omission	pepi.com	206.220.201.245 NS:ns.net10.net MX:pepi.com.mx1.net10.rcimx.net
Omission	pesi.com	216.56.243.144 NS:ns-1311.awsdns-35.org MX:d140747a.ess.barracudanetworks.com
Omission	peps.com	213.1.249.100 NS:ns.planit.com
Omission	ppsi.com	208.91.197.128 NS:ns33.worldnic.com MX:mx1.netsolmail.net
Omission	epsi.com	216.198.200.146 NS:ns23.domaincontrol.com MX:mail.epsi.com

Repetition	pepssi.com	-
Repetition	peppsi.com	172.98.192.36 NS:ns1.rentondc.com
Repetition	peepsi.com	23.20.239.12 NS:ns1.namebrightdns.com
Repetition	ppepsi.com	50.63.202.54 NS:ns01.domaincontrol.com MX:mailstore1.secureserver.net
Replacement	pemsi.com	-
Replacement	p3psi.com	184.168.131.241 NS:ns53.domaincontrol.com MX:mailstore1.secureserver.net
Replacement	peosi.com	192.185.188.86 NS:ns825.websitewelcome.com MX:mail.peosi.com
Replacement	peps9.com	-
Replacement	peps8.com	-
Replacement	pelsi.com	64.118.87.10 NS:ns1.mywwwserver.com MX:pelsi.com
Replacement	pepei.com	72.52.179.174 NS:ns1.parklogic.com
Replacement	pwpsi.com	23.20.239.12 NS:ns1.namebrightdns.com
Replacement	prpsi.com	-
Replacement	pspsi.com	NS:ns1.bdm.microsoftonline.com MX:pspsi-com.mail.protection.outlook.com
Replacement	pepsu.com	46.30.215.63 2a02:2350:5:104:cfc0:0:8428 NS:ns01.one.com MX:mx1.pub.mailpod6-cph3.one.com
Replacement	pepso.com	50.63.202.54 NS:ns63.domaincontrol.com MX:mailstore1.secureserver.net
Replacement	pepsj.com	192.232.223.72 NS:ns6175.hostgator.com MX:mail.pepsj.com
Replacement	pepdi.com	107.180.21.19 NS:ns67.domaincontrol.com MX:mail.pepdi.com
Replacement	pepai.com	185.53.179.24 NS:ns1.parkingcrew.net MX:mail.h-email.net
Replacement	pepyi.com	198.1.175.40 NS:ns1.dnsowl.com
Replacement	mepsi.com	192.64.147.150 NS:dns1.yoho.com MX:mx1.mepsi.com
Vowel-swap	pipsi.com	199.59.242.151 NS:ns1.bodis.com MX:mx76.m2bp.com
Vowel-swap	popsi.com	35.186.238.101 NS:ns1.namefind.com
Vowel-swap	pepse.com	198.60.86.20 NS:ns1.markmonitor.com MX:mailn.scientech.com

Now let's run it again without the `-r` switch to find potential domains that we can register as phishing targets:

```
root@OSINT: python dnstwist.py pepsi.com
```

Here is a truncated list of some potential domains we can register to try to phish pepsi (if we had their permission):

Homoglyph	pepsi.com	-
Homoglyph	pepsi.com	-
Homoglyph	pepsi.com	-

Homoglyph	peṑṑsi.com	-
Homoglyph	peṑsī.com	-
Homoglyph	pěṑsi.com	-
Homoglyph	pěṑsi.com	-
Homoglyph	pěṑsī.com	-
Homoglyph	peṑsī.com	-
Homoglyph	peṑsi.com	-
Homoglyph	peṑsī.com	-
Homoglyph	ḅeṑsi.com	-
Homoglyph	ḅeṑsi.com	-
Homoglyph	peṑṑsi.com	-
Homoglyph	peṑsī.com	-
Homoglyph	ḅeṑsi.com	-
Homoglyph	peṑsī.com	-
Homoglyph	ḅeṑsi.com	-
Homoglyph	péṑsi.com	-
Homoglyph	peṑsī.com	-
Homoglyph	ḅeṑsi.com	-
Insertion	peḷṑsi.com	-
Insertion	peṑwsi.com	-
Insertion	p4eṑsi.com	-
Insertion	peṑsai.com	-
Insertion	pseṑsi.com	-
Insertion	peṑswi.com	-
Insertion	peṑasi.com	-
Insertion	peṑdsi.com	-
Insertion	pe4ṑsi.com	-
Insertion	peṑ0si.com	-
Insertion	peoṑsi.com	-
Insertion	pzeṑsi.com	-
Insertion	pe3ṑsi.com	-
Insertion	pezṑsi.com	-
Insertion	peṑsyi.com	-
Insertion	peṑxsi.com	-
Insertion	peṑpsi.com	-
Insertion	pewṑsi.com	-

Summary

This chapter covered a lot of ground in the realm of network and subdomain discovery and exploration. We looked at tools like Sublist3r, fierce, and Enumall, which are all essential to any digital investigator or penetration tester because of their ability to sniff out subdomains within a target domain.

This chapter also touched on services like Shodan and Censys.io, which are big repositories of discoverable server information, and can also be used to identify servers or devices for your target organization.

Finally, this chapter compared the search performance of these different tools and services by looking at the type of results they provide when testing websites of different organizational sizes.

At the risk of sounding redundant, you should never rely on a single tool for all of your results.

The next chapter will build on these results by taking our newly discovered subdomains and looking for network activity, and ultimately information spillage across an organization's infrastructure.

Looking for Network Activity (Advanced NMAP Techniques)

This chapter focuses on identifying active hosts and ports using advanced NMAP scanning techniques. For almost every engagement, NMAP is the holy grail of “which tool to use first” to look for active hosts.

Because NMAP is such an unavoidable tool in every hacker’s and investigator’s arsenal, I feel it has already been covered to death in a multitude of other books. Rather than rehash the basics of performing NMAP scans, this chapter will focus on several advanced use cases and techniques that I have used to discover host activity, even through the most challenging firewalls.

Getting Started

A significant amount of critical data can be derived from performing NMAP scans, which can really set the tone for the rest of your engagement. Before we get started, this book assumes you have a basic understanding of NMAP—what it is, how it works, and how to use it. If you are new to NMAP, you might want to check out some free online tutorials; or if you’re like me, just play around with it.

Preparing a List of Active Hosts

This is typically the first scan I run. It is designed to scan all of the local IP space and come back with a nicely formatted list of which hosts are active. I use this for internal penetration tests, but it can easily be adapted to find which external addresses are live.

I use the following command to get a list of all active hosts on an internal network:

```
root@OSINT: nmap -n -sn 10.0.0.0/8 172.16.0.0/12 192.0.0.0/16 -oG - | \
awk '/Up$/{print $2}' > outputfile.txt
```

This command does the following:

1. `-sn` performs a ping scan (use `-ss` if pings are disabled)
2. Scans all internal IP spaces
3. `-oG -` outputs to greppable format
4. Sends output to AWK to only display IPs that are flagged as Up
5. Sends output to `outputfile.txt`

The output will be a very clean list of *active* IP addresses that you can use for your next task. The list will look like this:

```
192.168.0.1
192.168.0.2
192.168.0.3
... and so on.
```

Full Port Scans Using Different Scan Types

I rarely have luck with one scan type over another. It is usually luck of the draw, and I always find that it is good practice to run a complete NMAP scan multiple ways to make sure I haven't missed anything. *I also scan every port. Always.*

For less sophisticated networks, going in with a typical SYN scan works because the target machine will respond with a SYN/ACK if the port is open and a RST (reset) if it is closed.

But in reality, standard TCP scans using SYN or FIN will typically get blocked, especially if you are dealing with any sort of IDS /IPS. A good place to start is with an Xmas scan (`-sX`) but if you are not getting the results you would like, you should experiment with `-scanflags`.

This is how we configure our scan:

- `-p-`: Scan all ports, rather than just the default or standard ones. This will add some time to your scan as NMAP will be scanning ports 1-65535.
- `-sU`: Scan UDP.

Most scans only focus on TCP ports. Several services and ports use UDP to communicate and are often missed. Our previous scan types (`-sS`, `-sX`, etc.) will not find UDP ports. To find these ports and services, we need to include the `-sU` switch.

- `--scanflags`: This will tell NMAP which scan types to use. When running an NMAP scan, you have to define this setting. Some scan types include null scan, FIN scan, and Xmas scan. `--scanflags` allows you to set multiple scan types.
- Settings worth trying are `PSH`, `FINPSH`, and `SYNFIN`. `PSH` will try to get a response by manipulating the `PSH` flag of a TCP header. `FINPSH` will try the same tactic but will also try manipulating the `FIN` flags. `SYNFIN` will try manipulating the `FIN` flags and will also try to get a response using the `SYN/ACK` method.

Let's run a full port scan of `pepsi.com`:

```
root@OSINT: nmap -v -p- -sX -sU pepsi.com --scanflags PSH
```

The results of this scan were not ideal. All ports are coming back as open/filtered:

```
Starting Nmap 7.70 ( https://nmap.org ) at 2018-12-11 01:03 CST
Initiating Ping Scan at 01:03
Scanning pepsi.com (18.214.229.99) [4 ports]
Completed Ping Scan at 01:03, 0.06s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 01:03
Completed Parallel DNS resolution of 1 host. at 01:03, 0.00s elapsed
Initiating XMAS Scan at 01:03
Scanning pepsi.com (18.214.229.99) [65535 ports]
XMAS Scan Timing: About 18.48% done; ETC: 01:06 (0:02:17 remaining)
Stats: 0:00:39 elapsed; 0 hosts completed (1 up), 1 undergoing XMAS Scan
XMAS Scan Timing: About 25.85% done; ETC: 01:06 (0:01:52 remaining)
XMAS Scan Timing: About 53.74% done; ETC: 01:06 (0:00:59 remaining)
Completed XMAS Scan at 01:05, 117.00s elapsed (65535 total ports)
Initiating UDP Scan at 01:05
Scanning pepsi.com (18.214.229.99) [65535 ports]
Stats: 0:02:17 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 11.73% done; ETC: 01:08 (0:02:30 remaining)
UDP Scan Timing: About 35.76% done; ETC: 01:08 (0:01:30 remaining)
UDP Scan Timing: About 66.45% done; ETC: 01:07 (0:00:40 remaining)
Completed UDP Scan at 01:07, 108.80s elapsed (65535 total ports)
Nmap scan report for pepsi.com (18.214.229.99)
Host is up (0.058s latency).
rDNS record for 18.214.229.99: ec2-18-214-229-99.compute-1.amazonaws.com
All 131070 scanned ports on pepsi.com (18.214.229.99) are open|filtered
```

TCP Window Scan

The `-sW` switch of NMAP will perform a “window scan.” A TCP window scan is similar to an ACK scan but has been designed to differentiate between open and closed ports instead of showing the ports as unfiltered:

```
root@OSINT: nmap -v -p- pepsi.com -sW
```

```
Starting Nmap 7.70 ( https://nmap.org ) at 2018-12-11 01:15 CST
Initiating Ping Scan at 01:15
Scanning pepsi.com (18.214.229.99) [4 ports]
Completed Ping Scan at 01:15, 0.05s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 01:15
Completed Parallel DNS resolution of 1 host. at 01:15, 0.00s elapsed
Initiating Window Scan at 01:15
Scanning pepsi.com (18.214.229.99) [65535 ports]
Window Scan Timing: About 13.16% done; ETC: 01:19 (0:03:25 remaining)
Window Scan Timing: About 26.85% done; ETC: 01:19 (0:02:46 remaining)
Window Scan Timing: About 41.15% done; ETC: 01:19 (0:02:10 remaining)
Window Scan Timing: About 54.07% done; ETC: 01:19 (0:01:43 remaining)
Window Scan Timing: About 64.15% done; ETC: 01:19 (0:01:24 remaining)
Window Scan Timing: About 74.69% done; ETC: 01:19 (0:01:01 remaining)
Window Scan Timing: About 85.98% done; ETC: 01:19 (0:00:34 remaining)
Stats: 0:03:37 elapsed; 0 hosts completed (1 up), 1 undergoing Scan
Window Scan Timing: About 89.93% done; ETC: 01:19 (0:00:24 remaining)
Completed Window Scan at 01:19, 244.88s elapsed (65535 total ports)
Nmap scan report for pepsi.com (18.214.229.99)
Host is up (0.061s latency).
rDNS record for 18.214.229.99: ec2-18-214-229-99.compute-1.amazonaws.com
Not shown: 65533 filtered ports
PORT      STATE SERVICE
80/tcp    closed http
443/tcp   closed https
```

Look at that! We can at least see that ports 80 and 443 are open (which we already knew, but nice to have confirmation).

Working against Firewalls and IDS

Any modern firewall or intrusion detection system will come with a huge list of preconfigured block rules intended to block or minimize your ability to scan their infrastructure. If the traffic is not being blocked outright by a firewall, there is a possibility that it is being monitored by an IDS. This section provides helpful guides to help circumvent detection by these systems.

Using Reason Response

When NMAP runs a port scan, it returns a value of each port as being in a state of either open, closed, or filtered, and also provides the name of the service typically running on that port. The `--reason` switch will also return the reason why NMAP has determined a port to be in a particular state. In other words, if a port is showing as online, NMAP will also return the reason code that it used to determine why the port is online.

Identifying Live Servers

Have you ever been in a situation where you are scanning a network range and *every* port on *every* IP shows active? This is not a fun situation to run into, especially when you're not expecting it. Some modern IDS/IPS can detect that you are performing an NMAP scan and switch its responses to show every port as open. Not knowing this, you set your NMAP command to run and output to an XML file, then walk away figuring it will take a few hours. When you come back, you find the XML has grown to multiple GBs in size because NMAP is logging every port as open. It happens.

There are two ways to get around this. The first is by using the reason response codes. The second is with timing attacks (which will be discussed in the next section). Reason response codes can be used to detect live servers because of the way in which an active (or nonexistent) server responds to your probe requests.

When using the `--reason` switch, you will receive one of three responses:

- Closed: `reset`
- Open: `syn-ack`
- Filtered: `port-unreach`

If the port is genuinely unreachable, there won't be a response and the result will come back as filtered. However, if you receive a `reset` response, you know there is something live on the other end.

Building on our previous example, here is a FIN scan of a list of /24 (real IPs omitted). This is an actual scenario where a client owned the entire range and every IP is designed to display fully active ports to throw off an attacker:

```
root@OSINT: nmap -T4 -sF --send-ip --reason 1.2.3.4/24 -oX new-out.xml
```

The following XML is a sample of the results. Every IP shows a status of "up," but notice the value of the `reason` field. In cases where there was no activity, the reason is "no-responses". In cases where a firewall is legitimately blocking our scans, the response will show "resets":

```
<host starttime="1544321477" endtime="1544321651"><status state="up"
reason="echo-reply" reason_ttl="246"/>
```

```
<address addr="1.2.3.4" addrtype="ipv4"/>
<hostnames>
</hostnames>
<ports><extraports state="closed" count="1000">
<extrareasons reason="resets" count="1000"/>
</extraports>
</ports>
<times srttp="39452" rttvar="16489" to="105408"/>
</host>
<host starttime="1544321477" endtime="1544321604"><status state="up"
reason="echo-reply" reason_ttl="53"/>
<address addr="1.2.3.5" addrtype="ipv4"/>
<hostnames>
<hostname name="cassutility.com" type="PTR"/>
</hostnames>
<ports><extraports state="open|filtered" count="1000">
<extrareasons reason="no-responses" count="1000"/>
</extraports>
</ports>
<times srttp="23131" rttvar="23131" to="115655"/>
</host>
<host starttime="1544321477" endtime="1544321639"><status state="up"
reason="echo-reply" reason_ttl="53"/>
<address addr="1.2.3.6" addrtype="ipv4"/>
<hostnames>
</hostnames>
<ports><extraports state="open|filtered" count="1000">
<extrareasons reason="no-responses" count="1000"/>
</extraports>
</ports>
<times srttp="23763" rttvar="23763" to="118815"/>
</host>
<times srttp="22516" rttvar="22516" to="112580"/>
</host>
<host starttime="1544321477" endtime="1544321633"><status state="up"
reason="echo-reply" reason_ttl="53"/>
<address addr="1.2.3.7" addrtype="ipv4"/>
<hostnames>
</hostnames>
<ports><extraports state="open|filtered" count="1000">
<extrareasons reason="no-responses" count="1000"/>
</extraports>
</ports>
<times srttp="22949" rttvar="22949" to="114745"/>
</host>
```

Using this information, you can filter for IPs with a “resets” response in order to determine an actual list of live addresses.

Firewall Evasion

Firewall evasion using NMAP is becoming increasingly difficult as IDS/IPS are becoming more sophisticated. There are still some gotchas and many instances where a lot of these devices are misconfigured.

Some newer IDS configurations will show that all ports are closed. This is obvious when you are testing an IP from a website since the port of the website should at least be showing as open.

Distributed Scanning with Proxies and TOR

One way to completely anonymize your scans is to distribute your scans across random proxies. The following sections describe a few ways to accomplish this.

Rotating Proxy Service

There are a lot of services out there that will sell you rotating proxies (just Google “rotating proxies”). The way it works is that you get a single IP to connect through, and that IP connects out to the Internet using any number of randomly rotating proxies. Now you can easily distribute your NMAP scans across thousands of different IP addresses, which can have a drastically different impact on your scan results.

To use a proxy with NMAP, specify the address using the `--proxy` switch:

```
root@OSINT: nmap -sX -proxy http://1.1.1.1:1080 -iL targetlist.txt
```

Proxy Chains

Rather than using a single proxy, `proxychains` allows the ability to string together multiple proxies, making it harder to detect the source IP address. Once `proxychains` is installed on your Linux system, you can edit the `proxychains.conf` file to add a list of proxies that you want to chain together. The result is similar to how TOR distributes your traffic, but TOR is pretty slow. This approach is much faster.

Running `proxychains` is fairly simple. The command is:

```
proxychains <the command you want to run>
```

Running an NMAP command through `proxychains` would look like this:

```
root@OSINT: proxychains nmap -T4 -sX 1.2.3.4
```

PRO TIP: WILLIAM MARTIN

Another way to achieve a similar effect is using TOR proxies. You can already use a proxy in NMAP using the `--proxy` switch, and send the scan through a local TOR proxy. Or you can use `proxychains` to string together multiple proxies instead of

using the TOR network. There are also great services like Storm Proxies that will do this for you for a monthly price. This approach is best if you do not want to maintain your own proxy network. Storm Proxies automatically rotates 70,000 proxies for you so you don't have to worry about having your IP detected or the hassle of setting up and running your own TOR proxy.

Fragmented Packets/MTU

Although there is a very low likelihood that just using the fragmented packets switch will work (-f), it's worth trying in case there is a misconfigured IDS.

The -f switch tells NMAP to send 8-byte packets, fragmenting the probe into much smaller packets:

```
root@OSINT: nmap -f -sX -Pn -v pepsi.com
```

The --mtu (maximum transmission unit) option is similar to -f in that it sends a limited amount of data during the transmission. This can potentially confuse some older firewalls and will take some experimentation to get working properly:

```
root@OSINT: nmap -sX -v --mtu 32 pepsi.com
```

Service Detection Trick

If a port is showing as closed or filtered, you might be able to trick the host into providing a response by querying the service version. The trick behind both of these techniques is to force NMAP to assume that all ports are open with the -Pn switch. Without needing to decide if a port is open or closed, NMAP will run all of its operating system and service detection tests regardless. You might get lucky and receive a response.

The -A switch will enable “advanced and aggressive” features of OS detection. Using the -A switch is the same as specifying a few different NMAP options like -O (OS detection), and -sC (which will run the default scripts against a host).

The -sV switch will enable version detection on a port. This is incredibly noisy due to the scan having to test for multiple versions. If you don't care about making noise, this might be worth trying. The -sV switch is often needed to differentiate between open and filtered UDP ports. Also important to note, version detection is very slow as it involves sending a large number of application-specific probes to every open port. Because of that, this command also has the ability to crash poorly written applications, so use with caution.

The following results demonstrate how this works. First, let's run a standard Xmas scan against `pepsi.com`:

```
root@OSINT: nmap -sX pepsi.com
Starting Nmap 7.70 ( https://nmap.org ) at 2019-03-08 23:45 CST
Note: Host seems down. If it is really up,
but blocking our ping probes, try -Pn
Nmap done: 1 IP address (0 hosts up) scanned in 3.07 seconds
```

Now let's run a scan against the same host, but instruct NMAP to assume all ports are open and to perform a service version detection:

```
root@OSINT: nmap -Pn -sV pepsi.com
Starting Nmap 7.70 ( https://nmap.org ) at 2019-03-08 23:29 CST
Nmap scan report for pepsi.com (45.60.135.51)
Host is up (0.049s latency).
Other addresses for pepsi.com (not scanned): 45.60.75.51
Not shown: 778 filtered ports
```

PORT	STATE	SERVICE	VERSION
25/tcp	open	http	Incapsula CDN httpd
53/tcp	open	domain?	
80/tcp	open	http	Incapsula CDN httpd
81/tcp	open	http	Incapsula CDN httpd
82/tcp	open	http	Incapsula CDN httpd
83/tcp	open	http	Incapsula CDN httpd
84/tcp	open	http	Incapsula CDN httpd
85/tcp	open	http	Incapsula CDN httpd
88/tcp	open	http	Incapsula CDN httpd
89/tcp	open	http	Incapsula CDN httpd
90/tcp	open	http	Incapsula CDN httpd
99/tcp	open	http	Incapsula CDN httpd
389/tcp	open	ssl/http	Incapsula CDN httpd
443/tcp	open	ssl/http	Incapsula CDN httpd
444/tcp	open	ssl/http	Incapsula CDN httpd
555/tcp	open	http	Incapsula CDN httpd
587/tcp	open	http	Incapsula CDN httpd
631/tcp	open	http	Incapsula CDN httpd
636/tcp	open	ssl/http	Incapsula CDN httpd
777/tcp	open	http	Incapsula CDN httpd
800/tcp	open	http	Incapsula CDN httpd
801/tcp	open	http	Incapsula CDN httpd
843/tcp	open	http	Incapsula CDN httpd
888/tcp	open	http	Incapsula CDN httpd

```
[results truncated]
1 service unrecognized despite returning data.
SF-Port53-TCP:V=7.70%I=7%D=3/8%Time=5C834F5B [truncated]
```

Now that we know the host is behind a CDN, we can probably assume that most of these ports are not actually open. Regardless, these results are significantly different than the first scan.

Low and Slow

The `-T` switch of NMAP will change the timing between requests. Timing options can be used to fool advanced detection systems. If you spread out your scans, it may appear less likely that you are actually performing a port scan. Lowering the speed between requests is a good way to avoid firewall detection, but the cost of that is dramatically increasing your scan time. The following timing settings are available:

- T0: Paranoid (waits 5 minutes between sending each probe, not generally detectable by IDS/IPS)
- T1: Sneaky (waits 15 seconds)
- T2: Polite
- T3: Normal
- T4: Aggressive
- T5: Insane (easily detectable)

In a typical scan situation, I often use `-T4`. `-T3` is NMAP's default behavior, so `-T4` will give you a bit of a speed boost. Using the timing switch will not appear any different in the tool output but may yield different results. The following scan example sets the timing between scans to "aggressive," which sets a maximum delay of 10ms between each probe:

```
root@OSINT: nmap -sX -T4 yourtarget.com
```

Bad Checksums, Decoy, and Random Data

This section will look at three different NMAP techniques that involve sending invalid or spoofed data to the target in order to elicit a response.

Bad Checksums

The TCP/IP protocol uses "checksums" to ensure data integrity.

By crafting packets with incorrect checksum information, we might be able to trick the target host into sending a response. This typically only works in the case of misconfigured servers, but if you are stuck it is worth trying, especially if you are trying to evade a firewall.

We can send bad checksums to our targets using the `-badsum` switch. Because of the CDN, we saw in the earlier scan that the results of using `--badsum` can be misleading. When using `--badsum` against `pepsi.com`, we see the following results:

```
root@OSINT: nmap -sX -T4 --badsum 45.60.135.51
Starting Nmap 7.70 ( https://nmap.org )
Nmap scan report for 45.60.135.51
```

```
Host is up (0.050s latency).
Not shown: 778 filtered ports
PORT      STATE SERVICE
25/tcp    open  smtp
53/tcp    open  domain
80/tcp    open  http
81/tcp    open  hosts2-ns
82/tcp    open  xfer
83/tcp    open  mit-ml-dev
84/tcp    open  ctf
85/tcp    open  mit-ml-dev
88/tcp    open  kerberos-sec
89/tcp    open  su-mit-tg
90/tcp    open  dnsix
99/tcp    open  metagram
389/tcp   open  ldap
443/tcp   open  https
444/tcp   open  snpp
555/tcp   open  dsf
587/tcp   open  submission
```

The results of all the ports came back as open, even though we already know there is a CDN behind them that is impacting our results.

Decoy Scan

When performing a decoy scan, NMAP will spoof additional packets that appear to be coming from a number of different spoofed decoy addresses. This might make a host think that it is being bombarded from multiple systems simultaneously and cause the IDS/firewall to ignore the scans from our main system.

To include decoy packets, use the `-D` command, followed by `RND:[number]`. This will tell NMAP to generate X random decoys.

This command is more about hiding your IP than actually bypassing an IDS. When performing a decoy scan, an IDS might report 10 different port scans from unique IP addresses, not knowing which IP is actually performing the scan. This can be a good way to mask your actual IP address by creating a lot of noise. Obviously, this attack is very noisy, but if the goal is to mask the IP of the machine performing the scan, please note that your actual IP will still be in the logs. A better approach to mask your IP is to use proxies as previously discussed.

The following command will generate 10 decoy packets while using a SYN Scan against a target:

```
root@INTEL: nmap -D RND:10 -sS pepsi.com
```

The results are the same as you just saw since we are not changing the type of scan, only adding decoys.

Change Data Length

Port scanners typically send the same size requests, so some firewalls/IDS are designed to detect port scans by looking at the size of the packet. We can try to avoid that type of detection by using the `-data-length` parameter and specifying additional data to be sent with the packets. If you are going to experiment with this, try scanning a single port that you know is open (like 80 or 443) and play around with the packet size until you receive a response that you like:

```
root@OSINT: nmap -sS -data-length 300 scanme.nmap.org

Starting Nmap 7.01 ( https://nmap.org ) at 2018-12-12 08:52 UTC
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.085s latency).
Other addresses for scanme.nmap.org (not scanned):
2600:3c01::f03c:91ff:fe18:bb2f
Not shown: 996 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
9929/tcp  open  nping-echo
31337/tcp open  Elite

Nmap done: 1 IP address (1 host up) scanned in 2.41 seconds
```

IP/MAC Address Spoofing

Another method for trying to bypass firewall detection is to spoof the MAC address of your host. This technique can be very effective if the server has a MAC filtering rule that only allows traffic from certain MAC addresses. The downside is that you will need to know the MAC address of a machine that is allowed to query your servers. This technique is more likely to be used on an internal network scan. If you know a target uses a specific vendor, you can try to gather sample MAC addresses from their products and use those in your scan.

Similarly, you can spoof the source port or an IP address so the target thinks you are coming from a host within the target's subnet. There are instances where firewalls will allow requests from certain ports or IPs, so this might be worth trying.

NOTE Spoofing your IP address means you will *never* see the return packet unless you have access to that IP address. Spoofing an IP address of a scan requires you to bind your request to a network interface using the `-e` switch. This will typically require you to set up a fake internal network card on your computer (like `vnic1`) and

send the request from there. Spoofing a MAC address is a tactic not generally used on external discovery scans. It is most effective during internal network scans when you are able to identify trusted machines within a network.

The following command will spoof an IP address:

```
nmap -sF -e vnic1 -S 192.168.0.182 pepsi.com
```

To spoof a MAC address within a network, use the `--spoof-mac` switch:

```
nmap -sF -e vnic1 --spoof-mac <mac address> pepsi.com
```

Firewalking

Firewalking gathers information about a host by using an IP TTL (time to live) expiration algorithm to determine if a host is active or not. In a nutshell, the route to the host is determined using traceroute. A packet is then sent to the host with a TTL equal to the distance to the target. If the packet times out, it is resent with the previous value minus one. If an ICMP type 11 code 0 (TTL exceeded) is received, that means the packet was forwarded and the port is not blocked. If no response is received, the port is blocked on the gateway. Since traceroute is dependent on an IP layer (the TTL field), any transport protocol can be used the same way (TCP, UDP, and ICMP).

NMAP scripting engine (NSE) has a firewalk script that will automatically perform all of these steps. The script will try to discover the firewall rules using the IP TTL expiration method previously described.

We can run the firewalk script using the following command:

```
nmap --script=firewalk --script-args=firewalk.max.probed.ports=5 \
--traceroute host
```

Comparing Results

In the previous examples, we've run a number of different scan types and presumably output them all to some sort of format that we can go back to and look at (I usually use XML).

Ndiff is a tool similar to the Linux `diff` command. It will show you differences between files. Ndiff is specifically designed to show you differences between NMAP results.

I typically run three or four different scan variations to make sure I have all of my bases covered. Ndiff is designed in these cases to essentially show you what has changed between the different scans.

Another use case would be a sysadmin that runs an NMAP scan on his network every week and wants to see if anything new has popped up. Ndiff would be the fastest way to do that.

Running Ndiff is extremely simple. Just run `ndiff` followed by the two files you want to compare:

```
root@OSINT: ndiff nmap-scan1.xml nmap-scan2.xml
```

Our output will show us any differences between the two scan types:

```
-Nmap 7.01 scan initiated as: nmap -p- -sX -iL ips.txt -oX nmap-scan1
+Nmap 7.01 scan initiated as: nmap -sP -T4 --send-ip --reason \
-iL ips.txt -oX nmap-scan2
```

```
1.2.3.4:
-Not shown: 65534 closed ports
  PORT      STATE      SERVICE VERSION
-11211/tcp  open|filtered
```

```
1.2.3.5:
-Not shown: 65535 open|filtered ports
```

```
1.2.3.6:
-Not shown: 65534 open|filtered ports
  PORT    STATE  SERVICE VERSION
-443/tcp  closed https
```

```
1.2.3.7
-Not shown: 65534 open|filtered ports
  PORT    STATE  SERVICE VERSION
-443/tcp  closed https
```

```
1.2.3.8:
-Not shown: 65535 open|filtered ports
```

```
-1.2.3.9:
+hostname (1.2.3.9):
-Not shown: 65535 open|filtered ports
```

```
12.109.109.165:
-Not shown: 65535 open|filtered ports
```

```
1.2.3.10:
-Not shown: 65534 closed ports
  PORT      STATE      SERVICE VERSION
-11211/tcp  open|filtered
```

```
-1.2.3.11:
-Host is up.
-Not shown: 65535 open|filtered ports
```

```
1.2.3.12:
-Not shown: 65534 closed ports
  PORT      STATE      SERVICE VERSION
-11211/tcp  open|filtered
```


Based on these results, it looks like our Xmas scan has revealed that port 11211 is open on several hosts (which is typically associated with Memcached).

Styling NMAP Reports

One thing I have always had trouble with is visualizing NMAP results. There are a few programs that say they do this, but I have never found anything worthwhile. You can actually style the output of an NMAP XML file using XSL stylesheets with the `--stylesheet` parameter. Andreas Hontzia of Germany created a really nice XSL sheet to style NMAP output to a nicely readable report:

```
nmap -sS pepsu.com -oA filename --stylesheet \
https://raw.githubusercontent.com/honze-net/nmap-bootstrap-xsl/
master/nmap-bootstrap.xsl
```

Once the scan is complete, you can view the XML file in a web browser. The results will look like Figure 4.1.

Address	Port	Protocol	Service	Product	Version	CPE	More Info
45.60.75.51 - pepsu.com	25	tcp	http	Incopaula GDN httpd			
45.60.75.51 - pepsu.com	53	tcp	dns				
45.60.75.51 - pepsu.com	80	tcp	http	Incopaula GDN httpd			
45.60.75.51 - pepsu.com	81	tcp	http	Incopaula GDN httpd			
45.60.75.51 - pepsu.com	88	tcp	http	Incopaula GDN httpd			
45.60.75.51 - pepsu.com	83	tcp	http	Incopaula GDN httpd			
45.60.75.51 - pepsu.com	84	tcp	http	Incopaula GDN httpd			
45.60.75.51 - pepsu.com	85	tcp	http	Incopaula GDN httpd			
45.60.75.51 - pepsu.com	86	tcp	http	Incopaula GDN httpd			
45.60.75.51 - pepsu.com	89	tcp	http	Incopaula GDN httpd			

Showing 1 to 10 of 222 entries

Previous 1 2 3 4 5 ... 22 Next

Figure 4.1

Summary

This chapter covered a number of advanced techniques for using NMAP to identify activity on a target IP address. As firewalls and intrusion detection systems continue to become more sophisticated, different approaches should be utilized to ensure your scan results are accurate and complete. Now that we have looked at manual tools and approaches for detecting information on targets, the next chapter will focus on several automated tools that can be used to simplify our efforts and expand the volume of our search results.

Automated Tools for Network Discovery

We've looked at a number of manual command-line tools you can use to gain intelligence on targets, but what if you're in a rush or just want to take a shotgun approach to finding as much as you can all at once?

This chapter will look at three automated tools you can use to gather information quickly, and the various features of each. The tools in this section are specifically called out as “automated” because they dig deeper into any results by *automatically analyzing* any results and entities obtained in your first step to find new information.

For example, if an email address is discovered, the tools will automatically query various APIs to find as many connection points to that email address as possible; whereas with command-line tools, that is typically a manual process. These tools will automatically take your results from Step 1 and automate your next steps for you.

In the case of a discovered email address, the tool might look to see if it has been breached, or if there is any WHOIS data available from sources like SecurityTrails or Whoisology. In the case of a discovered IP address, the tools might automatically start looking up related domain names, or query threat intel feeds like VirusTotal to check if the IP is on any known blacklists.

The results between each tool are roughly 80% similar to each other. It's often that remaining 20% that can make the difference in an investigation and provide the most valuable (or unique) information.

In other words, there is no “one-stop shop” when it comes to intelligence gathering, which is why it is important to never rely on a single tool. Always run multiple tests to confirm that your results are accurate and complete.

The three tools discussed in this chapter are SpiderFoot, Intrigue.io, and Recon-NG.

A NOTE ABOUT ONE OF THE TEST TARGETS

For this chapter, we are going to look at a domain called `DualXCrypt.com` (Figure 5.1). The site offers a “decentralized smartphone” based on CopperheadOS. The domain was only online for a short time (you can still view it on archive.org). The interesting thing is that the people who designed the site advertised themselves as being a group of three well-known hackers from the “KickAss Forum.” I wouldn’t have known about this site if our friend Cyper (owner of the KickAss forum) didn’t post about it specifically stating that they were *not* related to KickAss. Seems like a pretty good reason to investigate.



Figure 5.1

SpiderFoot

SpiderFoot is an OSINT platform that’s based on the concept of information-chaining, meaning that one piece of collected data is used to automatically obtain the next piece of data in a recursive process until all possible OSINT for the target and related entities has been collected.

SpiderFoot chains and correlates data using more than 150 modules that query APIs, such as Shodan, VirusTotal, and Censys (to name a few). It even scrapes websites and performs port scans. SpiderFoot is an automated tool in the sense that it will recursively analyze any new information obtained through an initial scan in order to find as much information as possible. The results will exponentially balloon out very quickly.

Running SpiderFoot is pretty simple:

```
root@OSINT:/opt/spiderfoot: python sf.py 0.0.0.0:5001
```

From there, you can access SpiderFoot via a web browser on port 5001. You should take this time to set up your API keys in the app settings. Once you have that ready, running your first scan is pretty simple, as shown in Figure 5.2.

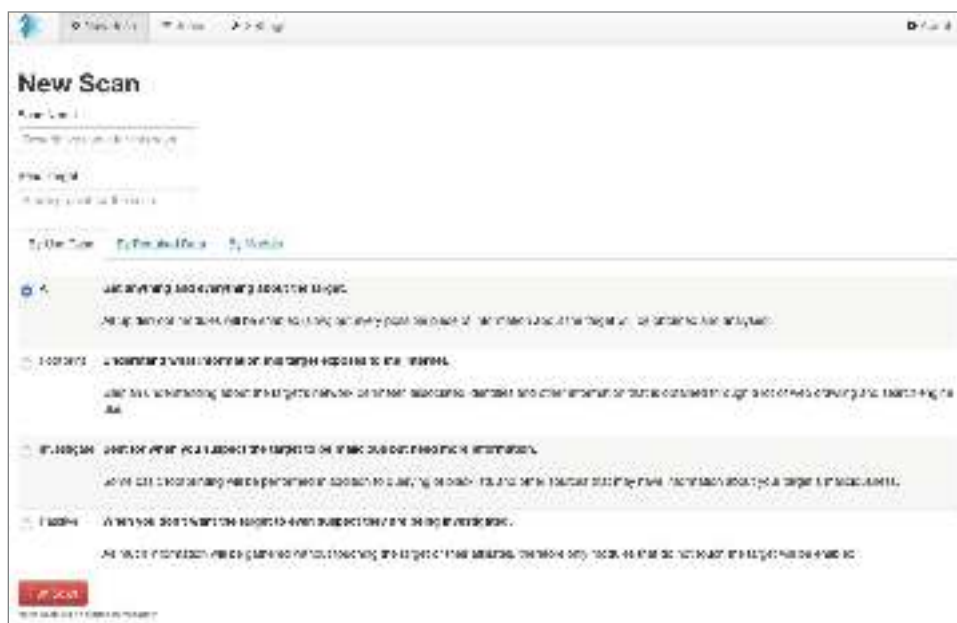


Figure 5.2

We have a few different scan options, but as mentioned earlier, we are going for the shotgun approach and will be scanning for everything to see what comes back.

The last known IP address of DualXCrypt was 185.165.169.124, so we are going to start there.

After we enter our target IP of 185.165.169.124, we click Run Scan and let SpiderFoot do its thing. When the scan is ready, we can look in the scans tab and see our initial results, as shown in Figure 5.3.

Name	Target	Started	Finished	Status	Elements	Action
185.165.169.124	185.165.169.124	2016-12-15 13:56:29	2016-12-15 13:56:59	FINISHED	112	  

Figure 5.3

Clicking the IP address will show us an expanded list of the different elements discovered by our scan (Figure 5.4).

185.165.169.124		
Status Browse Graph Scan Settings Log		
Type	Unique Data Elements	Total Data Elements
Affiliate - Email Address	79	101
BGP AS Membership	1	1
BGP AS Peer	3	3
Blacklisted IP on Same Subnet	11	11
Co-Hosted Site	47	49
Co-Hosted Site - Domain Name	40	47
Co-Hosted Site - Domain Whois	39	39
IP Address	1	1
Malicious Co-Hosted Site	9	9
Malicious IP Address	36	36
Malicious IP on Same Subnet	5	5
Netblock Membership	1	1
Open TCP Port	2	2
Search Engine's Web Content	2	2

Figure 5.4

We can see from the results in Figure 5.4 that there are a lot of really interesting avenues of new information to pursue. One item that stands out is the number of malicious co-hosted sites (9).

NOTE One of the reasons I don't often start with a shotgun approach is the sheer volume of returned results. Having to mentally process that much information in such a short amount of time makes it very easy to miss something. Unless you have a really solid system for cataloging and storing information, chances are something in these results will take you down one particular path, at which point it is very difficult to circle back and start over.

Expanding on that option (Figure 5.5), we can see the list of other malicious websites using the same IP address.

☐	VirusTotal [dnstars.vip] https://www.virustotal.com/en/domain/dnstars.vip/information/	dnstars.vip
☐	VirusTotal [dualxencrypt.com] https://www.virustotal.com/en/domain/dualxencrypt.com/information/	dualxencrypt.com
☐	VirusTotal [putlockerhd.cc] https://www.virustotal.com/en/domain/putlockerhd.cc/information/	putlockerhd.cc
☐	VirusTotal [tfile-search.cc] https://www.virustotal.com/en/domain/tfile-search.cc/information/	tfile-search.cc
☐	VirusTotal [tfile-video.org] https://www.virustotal.com/en/domain/tfile-video.org/information/	tfile-video.org
☐	VirusTotal [www.seventorrents.cc] https://www.virustotal.com/en/domain/www.seventorrents.cc/information/	www.seventorrents.cc

Figure 5.5

The results shown in Figure 5.5 are worth noting because having other malicious sites using the same IP address is unusual. However, given the context of the site and its owners, the discovery of other malicious sites using the same IP is not surprising. There could be similarities in domain owners worth exploring.

Looking at the other potentially malicious sites using the same IP may tell us about the target. Since we can see other potentially illegal sites like torrents and illegal video download sites, it also raises the question of where the site is being hosted.

Clicking the “dualxencrypt.com” link from SpiderFoot takes us directly to VirusTotal, where we can see what information they have on our target domain (Figure 5.6).



The screenshot shows the VirusTotal interface for the domain dualxcrypt.com. It displays a table of IP addresses associated with the domain over time, from November 2018 to January 2019. The table has columns for Date, IP Address, and a checkbox for selection.

Date	IP Address	
2018-11-20	185.165.169.124	☐
2018-10-25	25.328.20.43	☐
2018-09-25	71.39.237.210	☐
2018-09-25	75.106.210.155	☐
2018-09-27	21.80.233.23	☐
2018-09-28	21.80.224.180	☐
2018-09-28	21.80.225.179	☐
2018-09-18	21.111.25.141	☐
2018-09-18	21.19.150.148	☐
2018-09-18	21.11.77.279	☐

Figure 5.6

This is a pretty interesting find because this particular domain had a lot of IP movement over a one-month period. Each of these IPs is worth investigating because it is possible that one of them could be a personal web server, or even a home address. Before we do that, though, it might be worthwhile to see where this site is currently hosted.

A quick detour over to Censys.io (Figure 5.7) shows us the ISP is Flokinet, hosted in Seychelles. This information may or may not be useful in the future, but for now, it's worth noting.



The screenshot shows the Censys.io interface for the IP address 185.165.169.124. It displays basic information about the IP, including the network (FLOKINET), routing (185.165.169.0/24 via AS20061), and protocols (no publicly accessible services).

Basic Information	
Network	FLOKINET - SC (IS)
Routing	185.165.169.0/24 via AS20061
Protocols	no publicly accessible services

Figure 5.7

The WHOIS information shows registration with njalla.io, an anonymous domain registration service that also provides fully anonymous web hosting, so that's a dead end (Chapter 9 covers WHOIS in detail).

With nowhere left to explore with this particular avenue of information gathering, we can pivot and run subsequent scans on neighboring IPs or domain names, or perhaps we want to track down different leads.

NOTE When using Censys.io or Shodan, a search for “dualxcrypt.com” actually returns no results. In this case, [virustotal.com](https://www.virustotal.com) is the clear winner in terms of available information. However, plenty of situations exist in which the exact opposite is true. You should always test using multiple services.

There may not always be a wealth of information available. So let's try with a much more common target—a popular web forum, uberpeople.net.

Back on the SpiderFoot main menu, we can kick off a new scan for uberpeople.net, as shown in Figure 5.8.

Figure 5.8

Following our scan of uberpeople.net, we can already see that we have significantly more elements to work with (Figure 5.9).

☐	Name	Target	Status	Elements	Action
☐	uberpeople.net	uberpeople.net	FINISHED	1043	

Figure 5.9

Table 5.1 shows a list of the different data elements from our scan of uberpeople.net, which includes things like DNS records, email addresses, junk files, leaked site content (via pastebin), usernames, and more.

Table 5.1: List of Discovered Elements from SpiderFoot Scan

TYPE	UNIQUE DATA ELEMENTS
Account on External Site	12
Affiliate - Company Name	5
Affiliate - Domain Name	8
Affiliate - Domain Whois	7
Affiliate - Email Address	19
Affiliate - IP Address	20
Affiliate - Internet Name	13
BGP AS Membership	1
BGP AS Peer	9
Blacklisted Affiliate IP Address	1
Co-Hosted Site	2
Co-Hosted Site - Domain Name	1
DNS SPF Record	1
DNS TXT Record	2
Domain Name	1
Email Address	2
Email Gateway (DNS 'MX' Records)	1
Hosting Provider	2
IP Address	1
Internet Name	18
Internet Name - Unresolved	6
Junk File	36
Leak Site Content	1
Leak Site URL	2
Linked URL - Internal	287
Malicious Affiliate	1
Malicious Affiliate IP Address	10
Malicious Co-Hosted Site	2
Malicious IP Address	2
Malicious IP on Same Subnet	4
Name Server (DNS 'NS' Records)	2

TYPE	UNIQUE DATA ELEMENTS
Netblock Membership	1
Open TCP Port	18
Open TCP Port Banner	5
Raw DNS Records	15
SSL Certificate - Issued by	2
SSL Certificate - Issued to	4
SSL Certificate - Raw Data	8
SSL Certificate Host Mismatch	1
Search Engine's Web Content	60
Similar Domain	6
Similar Domain - Whois	5
Username	1

The total number of discovered records is 605. That is *a lot* of information to start with! It is almost too much information to take in all at once because of the many different avenues of research each different category could lead you down. If you haven't already, it is best to make sure you have a way of keeping all of your data cataloged and organized or you will absolutely miss and forget things along the way.

SpiderFoot HX (Premium)

SpiderFoot HX is the premium version of SpiderFoot. HX is purely cloud-based and contains a number of additional enhancements and capabilities including improved scan speeds, multi-threaded/concurrent scanning, better visualizations, team collaboration, risk identification, and significantly improved automated data correlations.

The upgraded automated correlation engine immediately caught my attention. In this section we will rescan the previous set of targets in SpiderFoot HX and provide a comparison of the differences.

To kick off the scans, Figure 5.10 shows the targets added to the New Scan window. All Modules has been selected as the scan profile, which shows the default new scan window with our targets.

Comparing the differences, we see right away that the HX version of SpiderFoot provides more results and will even classify those results by risk level. Figure 5.11 shows the initial risk results, with two high-risk items, two medium, and one low.

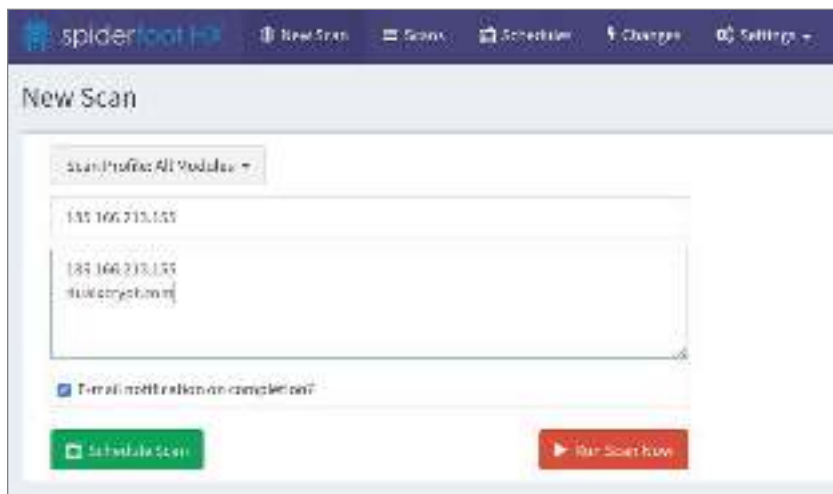


Figure 5.10



Figure 5.11

Digging into our results section, it is immediately obvious that the UI and graphical elements within HX are significantly improved from those found in the standard version of SpiderFoot, as shown in Figure 5.12.

TIP When working with executive clients, having graphical elements can often make the difference between a good report and an excellent report. I've found that most executives don't want a report loaded with a bunch of words that they most likely won't ever have the time to read. They want to see concise results, bullet points, and graphical elements they can use to easily present the findings to their board or executive team.

Overall, the results between the free and premium version of SpiderFoot are very similar. The majority of returned results (Figure 5.13) come back to the IP's subnet being on about 30 different blacklists. In addition, the HX version found an additional username associated with our target, a Reddit account called "dualxcrypt" (<https://www.reddit.com/user/dualxcrypt>). A great lead, but unfortunately the user has not posted anything.

Admittedly, this was a very difficult target with very little discoverable information. Let's see how the results for SpiderFoot HX compare to its open-source counterpart when scanning uberpeople.net, a much more popular target (Figure 5.14).

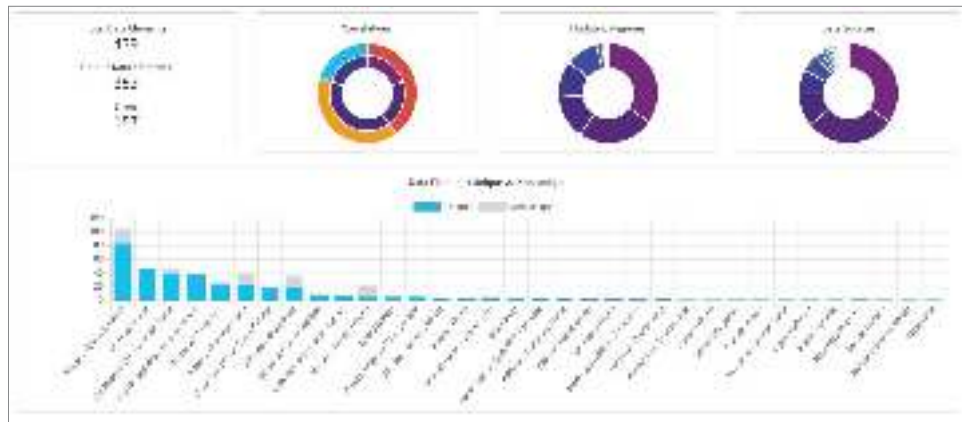


Figure 5.12

Correlation	Data Elements	Extrusability	Created
Entity is addressable via public IP (10.10.10.10.10.10.10.10)	19	CRITICAL	2018-12-17 17:06:13
Entity is addressable via IP (10.10.10.10.10.10.10.10)	19	CRITICAL	2018-12-17 17:06:13
Entity is addressable via IP (10.10.10.10.10.10.10.10)	19	CRITICAL	2018-12-17 17:06:13
Entity is addressable via IP (10.10.10.10.10.10.10.10)	19	CRITICAL	2018-12-17 17:06:13
Entity is addressable via IP (10.10.10.10.10.10.10.10)	19	CRITICAL	2018-12-17 17:06:13

Figure 5.13



Figure 5.14

The HX scans against `uberpeople.net` show 629 results with 1 critical and 2 medium-risk items (Figure 5.15).



Figure 5.15

Our earlier scans of `uberpeople.net` using the open-source version of Spider-Foot yielded 605 total results, while the improved HX version yielded 629 results. The 24 additional matches may not appear significant at first glance, but those numbers could easily be a result of many factors, including our sample test sites. Depending on your job function, the added visual components and ability to export usable graphical elements can make this a very worthwhile subscription upgrade.

Intrigue.io

Intrigue was developed by security developer and researcher Jonathan Cran, and you can download it at www.intrigue.io. Intrigue is the open-source engine that powers Intrigue.io, a platform for attack surface discovery. According to Cran, Intrigue Core was designed around the concept of graph-based discovery, allowing it to automate the process of attack surface discovery and security intelligence gathering.

The primary focus of Intrigue is to discover assets and “interesting things” about an organization from an external perspective. It is also becoming useful as a platform for dev-ops and security teams to pull security-related information together in a single pane of glass, allowing them to easily find connections between organizational entities.

Within Intrigue, every bit of information (or intelligence) is classified as an *entity*. Intrigue’s flexible data model uses tasks to perform functions that create these entities. Each entity can, in turn, be iterated upon with more tasks—either manually or automatically. Intrigue uses this data model, and another iteration concept called *machines*, to flesh out the assets and vulnerabilities of an organization.

What I find most impressive about Intrigue is its ability to process, detect, and automatically reprocess newly discovered entities.

For example, let’s say you are running a wide search on a domain name. Intrigue may bring back a number of entities including an IP address, related domain names, and subdomains (just to name a few). By using the “enrich” option, Intrigue will take each of those new entities and run an entire battery of tests on each them. The combinations and results quickly become exponential, saving you a significant amount of time in your initial discovery efforts.

Intrigue can perform a large number of tasks, including:

- Open database searches: Censys, SecurityTrails, and Shodan
- WHOIS lookups
- Public Trello board checks
- Bruteforcing of S3, Trello, and other public buckets
- URI spidering
- Directory and filename bruteforcing
- DNS/name server enumeration

- Certificate transparency searches (CRT.sh)
- Related nameserver and DNS record checks
- Related WHOIS domain checks
- FTP service discovery and data collection
- SSH, RDP, and VNC login (with automatic screenshots)
- NSEC Walking
- GitHub searching (with Gitrob integration)
- And more

AUTHOR COMMENT: JONATHAN CRAN

I find Intrigue Core to be incredibly particularly powerful at the application layer, due to the Ident library. Ident was designed specifically for application-layer fingerprinting and is kept current based on the ever-increasing scan data we have access to through Intrigue.io. This gives the core technology an ability to do version-specific fingerprinting and vulnerability matching.

With this many built-in tasks, as well as notifications and the concept of issues, Intrigue Core is more of a platform than a tool. The task and entity concept makes it easy to extend the app's functionality. And as we continue finding new assets, vulnerabilities, and misconfigurations, look for the platform to keep evolving.

To use Intrigue to kick off a basic automated query against a URI, click the Start button. Select Create Entity as the task, and set the type to URI (as shown in Figure 5.16).

Entities Tab

This is where the magic of Intrigue happens. The Entities tab displays a list of all discovered entity types. For `DualXCrypt.com`, we only see six types. For more popular targets (which we will see later in this section), this number will be much higher.

The following are the entity types for the discovered results of `DualXCrypt.com` (Figure 5.17):

- Domains: 3
- Uri: 1
- IPAddress: 1
- Netblock: 1

Start

Task: Create Entity

Entity Type: Url

Entity Name: http://www.dualxcrypt.com

Machine: Org Asset Discovery (Active)
Machine specifies the post-processor for each new entity.

Iterations: 4 iterations
Iterations specifies the depth to which the machine is run.

☒ Auto Enrich

Figure 5.16

Entity	details
• [IpAddress: 185.185.188.124]	• Synchronise
• [Domain: njallano]	• Add
• [Domain: njallio]	• Add
• [Domain: dualxcrypt.com]	• Add
• [NwifBlock: 185.185.188.0/22]	• SC-FLOCKNET-TO-20160828
• [Uri: http://www.dualxcrypt.com]	• Server Add Title

Figure 5.17

Clicking any of those entities will give us more options and transforms that we can run. For example, let's click the domain name. Figure 5.18 shows the entity detail page; on the right, we can see a list of options for new tasks we can run.

Clicking the task type shows a plethora of follow-on scans that can run on the domain (Figure 5.19), including searching CRT.SH (which we discuss in Chapter 10), checking public Google Groups leaks, Trello leaks, enumerating nameservers, Whoisology search, and much more. Each entity type has its own set of additional tasks that can be run to find additional information on your target.

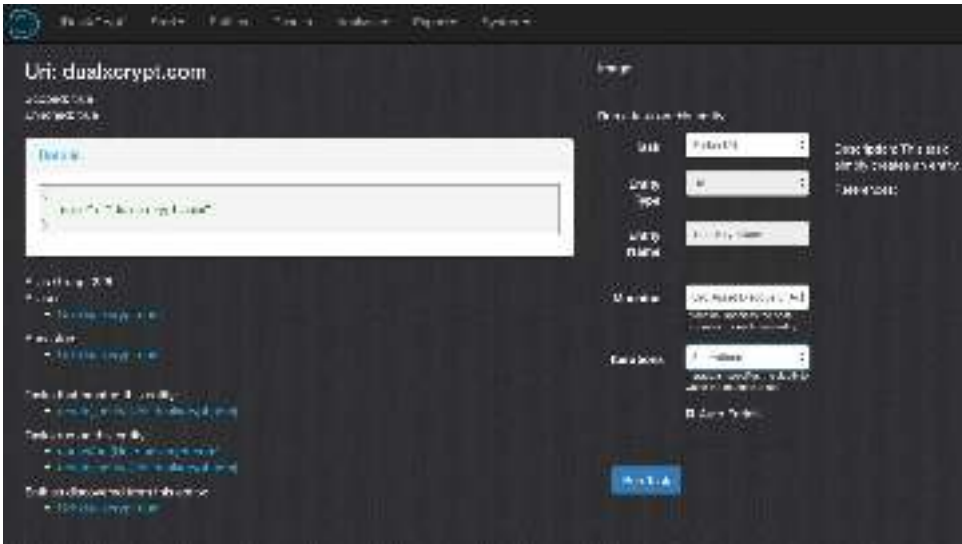


Figure 5.18



Figure 5.19

From the screen shown in Figure 5.20, we can also expand and see greater detail on the discovered entity. Clicking the IP address shows us general information about it, including its potential location. On the right, we can pull up our possible task list and run new actions based on this IP. Notice that the task types in Figure 5.20 are different than the previous task types. That's because the task types will change based on what can be performed with the given entity type.

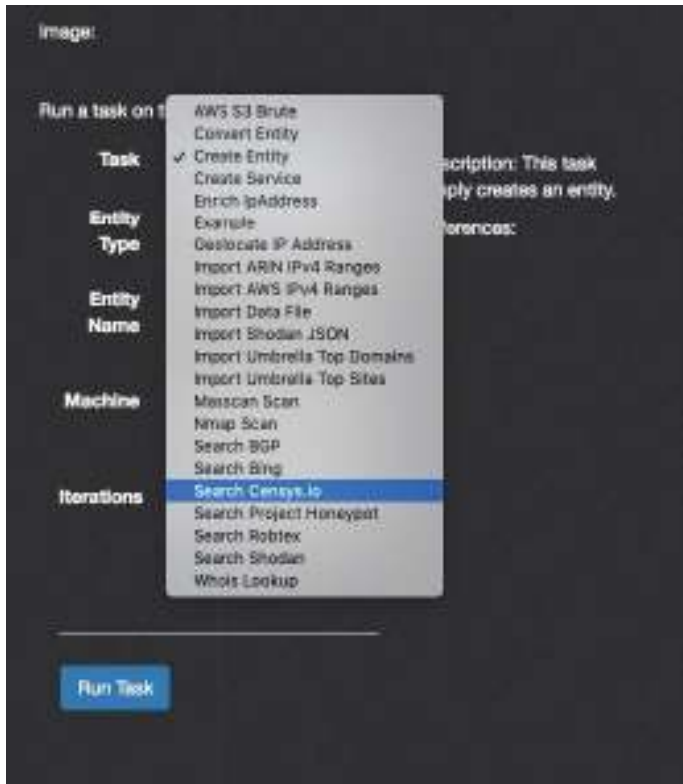


Figure 5.20

Analyzing *uberpeople.net*

Now that we have an understanding of the capabilities of Intrigue, let's restart the process to look at *uberpeople.net*. Figure 5.21 shows the Add Entity screen configured to kick off a scan for *uberpeople.net*.

Looking at the Results tab, we can see that a number of tasks have been automatically run on *uberpeople.net*.

Figure 5.22 shows a small sample of the different tasks that were automatically run, and the number of discovered entities per task.

Figure 5.21

The NMAP scan looks interesting. If we click that, we get not only a list of which ports were discovered but also a nice list of additional entities that we can explore further, as shown in the following code and Figure 5.23:

```
[_] Options: []
[_] Starting task run at 2018-12-18 03:52:42 UTC!
[_] Scan list is: ["185.165.169.124"], ports: 10
[_] Scanning 185.165.169.124 and storing in /tmp/nmap_scan_47907309.xml
[_] NMap options:
[_] Nmap Output:
```

Starting Nmap 7.01 (<https://nmap.org>) at 2018-12-18 03:52 UTC

Nmap scan report for 185.165.169.124

Host is up.

PORT	STATE	SERVICE	VERSION
21/tcp	filtered	ftp	
22/tcp	filtered	ssh	
23/tcp	filtered	telnet	
25/tcp	filtered	smtp	
80/tcp	filtered	http	
110/tcp	filtered	pop3	

```

139/tcp    filtered    netbios-ssn
443/tcp    filtered    https
445/tcp    filtered    microsoft-ds
3389/tcp   filtered    ms-wbt-server
53/udp     open|filtered domain
67/udp     open|filtered dhcp
123/udp     open|filtered ntp
135/udp     open|filtered msrpc
137/udp     open|filtered netbios-ns
138/udp     open|filtered netbios-dgm
161/udp     open|filtered snmp
445/udp     open|filtered microsoft-ds
631/udp     open|filtered ipp
1434/udp    open|filtered ms-sql-m

```

Too many fingerprints match this host to give specific OS details

I find Intrigue.io to be really unique not only because of the quantity of tasks/modules you can run but because of how wide you can go with those modules. For example, within the results shown in Figure 5.24, we can see webdisk.uberpeople.net and ftp.uberpeople.net.

dns_brute_sub on DnsRecord: ftp.uberpeople.net	0	complete
enrich_dns_record on DnsRecord: ftp.uberpeople.net	3	complete
dns_brute_sub on DnsRecord: www.uberpeople.net	0	complete
enrich_dns_record on DnsRecord: www.uberpeople.net	3	complete
dns_brute_sub on DnsRecord: opanel.uberpeople.net	0	complete
enrich_dns_record on DnsRecord: opanel.uberpeople.net	3	complete
dns_brute_sub on DnsRecord: www.mokotoband.uberpeople.net	0	complete
enrich_dns_record on DnsRecord: www.mokotoband.uberpeople.net	3	complete
dns_brute_sub on DnsRecord: mokotoband.uberpeople.net	3	complete
enrich_dns_record on DnsRecord: uberpeople.net	2	complete
enrich_dns_record on DnsRecord: mokotoband.uberpeople.net	3	complete
dns_brute_sub on DnsRecord: www.bedmaybedeakfast.uberpeople.net	0	complete
enrich_dns_record on DnsRecord: www.bedmaybedeakfast.uberpeople.net	3	complete
dns_brute_sub on DnsRecord: bedmaybedeakfast.uberpeople.net	3	complete
enrich_dns_record on DnsRecord: bedmaybedeakfast.uberpeople.net	3	complete
enrich_nameserver on DnsRecord: ns2.uberpeople.net	0	complete
enrich_nameserver on DnsRecord: ns1.uberpeople.net	0	complete
www_83_brute on DnsRecord: uberpeople.net	0	complete
public_google_groups_check on DnsRecord: uberpeople.net	0	complete
dns_brute_sub on DnsRecord: uberpeople.net	13	complete
search_crt on DnsRecord: uberpeople.net	4	complete
enumerate_nameservers on DnsRecord: uberpeople.net	2	complete
dns_brute_sub on DnsRecord: vas.uberpeople.net	0	complete
enrich_dns_record on DnsRecord: 194.154.10.116	0	complete
imap_scan on IpAddress: 194.154.76.235	39	complete
whole_backup on IpAddress: 194.154.76.235	1	complete
enrich_dns_record on DnsRecord: vas.uberpeople.net	3	complete
enrich_domain on DnsRecord: uberpeople.net	1	complete
enrich_ip_address on IpAddress: 194.154.76.235	2	complete
create_entity on DnsRecord: uberpeople.net	1	complete

Figure 5.22



Figure 5.23

Exploring the results for `webdisk.uberpeople.net`, we can see that the task options have changed and are specific to our new target. This time, we can run vulnerability checks, spider the host, take screenshots (useful when showing open RDP or VNC ports), and even perform a bruteforce against the target using commands like “URI Bruteforce” and “URI Bruteforce with Credentials.”

WARNING Bruteforcing credentials is outside the scope of this book and can be illegal if you do not have the right permissions. If you do use those tools, please use them legally and responsibly.

Analyzing the Results

Intrigue.io's results engine is incredibly impressive. Once your scans are complete, click the Entities tab to view everything it has collected.

For `uberpeople.net`, we can see the following discovered entities, shown in Figure 5.25:

- Total Entities: 58 entities
- NetworkService: 1
- DnsService: 1
- SshService: 1
- FtpService: 1
- SntpService: 1
- Uri: 34
- IpAddress: 1
- NetBlock: 1
- DnsRecord: 17

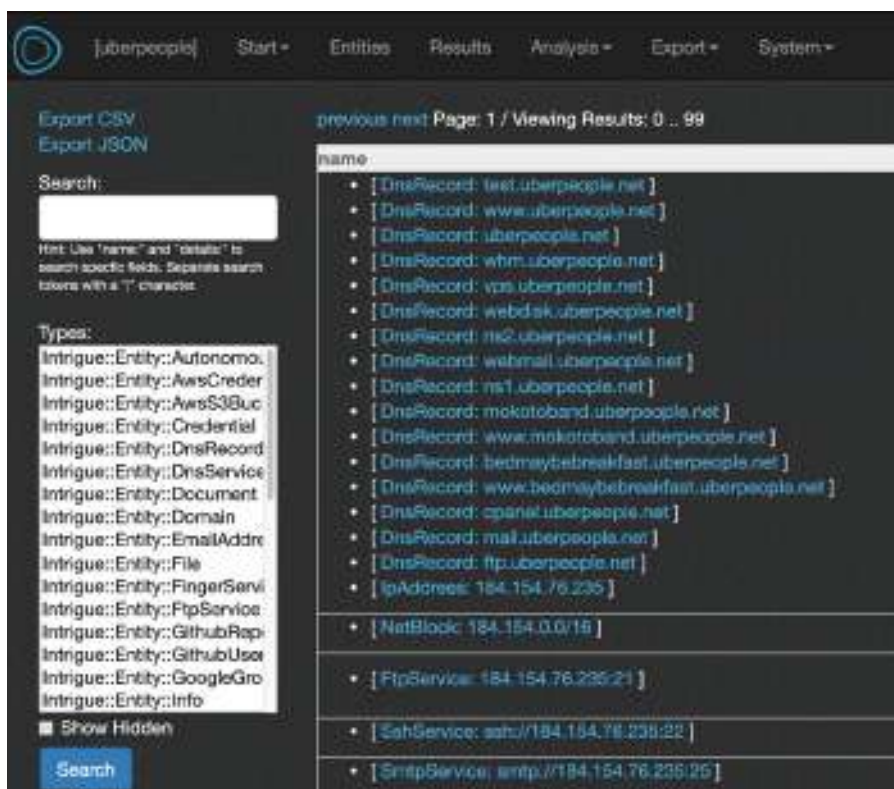


Figure 5.25

As the size of your targets increases, so will your results. To add some consistency to our testing, here are Intrigue's results when performing lookups on `pepsi.com`:

- Total Entities: 208 entities
- NetworkService: 3
- DnsService: 1
- SshService: 1
- Domain: 15
- Nameserver: 4
- SntpService: 1
- Uri: 72
- IpAddress: 20
- NetBlock: 10
- DnsRecord: 81

Working within the Entities section, each of the entities is listed in the master table. From here, you can click an entity to view additional information and run additional tasks.

You can also filter your list by clicking one of the Types and clicking the Search button. Doing so will display only the entity types you are looking for.

For example, let's filter our search by DNSRecords. To do this, click the entity type (DNSRecord) and then click Search. Figure 5.26 shows the updated screen with the filtered search results.

Exporting Your Results

Within the Entities page, you can choose to export your results in easy-to-use CSV or JSON formats. One other feature worth mentioning is Intrigue's graphing capability. You can choose to view your results in a maltego-style graph. This can be helpful for larger searches when looking for centralized connection points. To view a graph of your results, click the Analysis menu option. Figure 5.27 shows a graph generated by Intrigue.io.

Each of the nodes within the graph is clickable, so we can use this page to quickly access any of the entities and perform further lookups. Figure 5.28 shows a close-up of an interesting-sounding subdomain: `bedmaybebreakfast.uberpeople.net`.

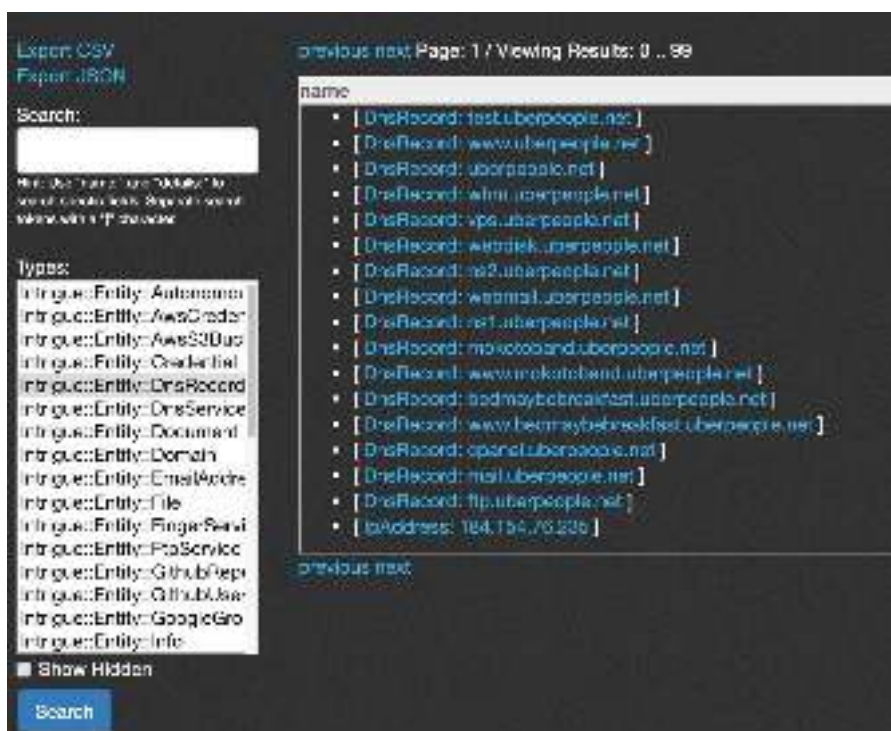


Figure 5.26

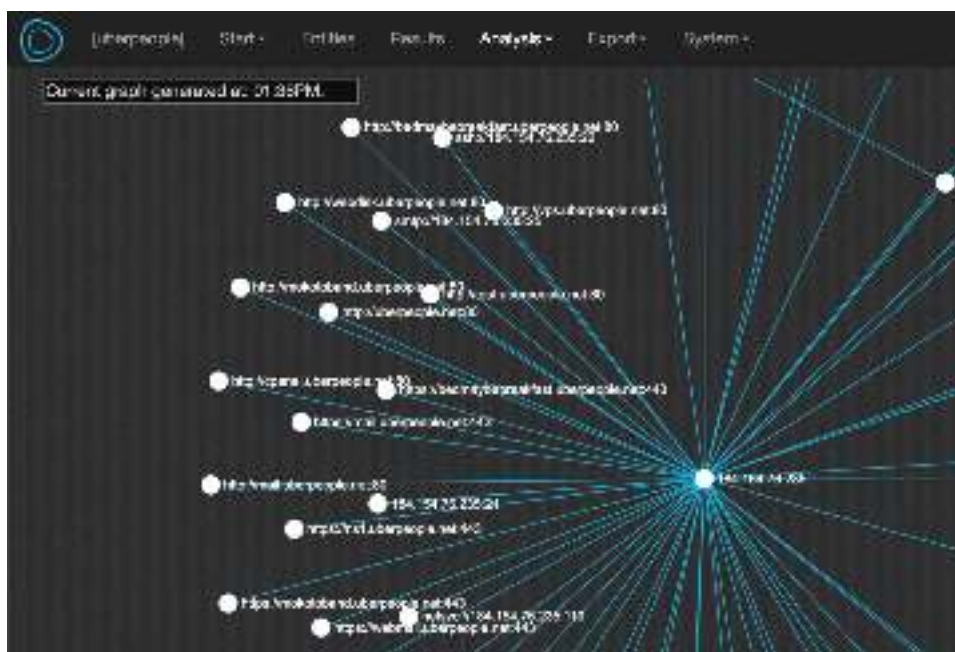


Figure 5.27



Figure 5.28

Clicking that subdomain entity will bring us to the entity detail page, shown in Figure 5.29.



Figure 5.29

This page will show any information discovered about a particular entity and will also allow us to run additional tasks such as performing a full URL spider and metadata extraction (which we discuss in greater detail in Chapter 12).

This should give you a pretty good idea of the capabilities of Intrigue. Now let's move on to our third and final tool of this chapter, Recon-NG.

Recon-NG

Recon-NG is probably one of the most widely used OSINT/recon gathering tools. It combines many of the same features we saw in the previous two tools within this chapter, but is purely command-line based. Sorry, no GUI available.

Recon-NG is very similar to Metasploit in that you have to manually load and run each module you want to use. The Recon-NG tool comes with many built-in modules organized by category, each with their own capabilities.

Recon-NG is not “automated” like the other tools, but it does have a lot of the same capabilities. Each module has its own set of commands and options. Since it is command-line based, you have to tell Recon-NG what you want it to do, and manually build upon each set of results.

Currently, 76 recon modules allow you to perform different types of searches on individual people, companies, websites, social media profiles, hosts, DNS, IPs, and more.

Running Recon-NG is a simple one-line command:

```
root@OSINT: ./recon-ng
```

After launching the application, you will be greeted with a Metasploit-esque page, shown in Figure 5.30.



Figure 5.30

From here, Recon-NG will expect your input on what to do next. If you are not sure which module to use, you can type `use` and press the Tab key to get a list similar to the following:

```
[recon-ng] [OSINT] > use

discovery/info_disclosure/cache_snoop
discovery/info_disclosure/interesting_files
exploitation/injection/command_injector
exploitation/injection/xpath_bruter
import/csv_file
import/list
```

```
recon/companies-contacts/bing_linkedin_cache
recon/companies-contacts/jigsaw/point_usage
recon/companies-contacts/jigsaw/purchase_contact
recon/domains-vulnerabilities/punkspider
recon/domains-vulnerabilities/xssed
recon/domains-vulnerabilities/xssposed
recon/hosts-hosts/bing_ip
recon/hosts-hosts/ipinfodb
recon/hosts-hosts/ipstack
recon/hosts-hosts/resolve
recon/hosts-hosts/ssltools
recon/hosts-hosts/reverse_resolve
recon/hosts-domains/migrate_hosts
recon/hosts-domains/migrate_
recon/companies-contacts/jigsaw/search_contacts
```

```
recon/companies-multi/github_miner
recon/companies-multi/whois_miner
recon/contacts-contacts/mailtester
recon/contacts-contacts/mangle
recon/contacts-contacts/unmangle
recon/contacts-credentials/hibp_breach
recon/contacts-credentials/hibp_paste
recon/contacts-domains/migrate_contacts
recon/contacts-profiles/fullcontact
recon/credentials-credentials/adobe
recon/credentials-credentials/bozocrack
recon/credentials-credentials/hashe.org
recon/domains-contacts/metacrawler
recon/domains-contacts/pgp_search
recon/domains-contacts/whois_pocs
recon/domains-credentials/pwnedlist/account_creds
recon/domains-credentials/pwnedlist/api_usage
recon/domains-credentials/pwnedlist/domain_creds
recon/domains-credentials/pwnedlist/domain_ispwned
recon/domains-credentials/pwnedlist/leak_lookup
recon/domains-credentials/pwnedlist/leaks_dump
recon/domains-domains/brute_suffix
recon/domains-hosts/bing_domain_api
recon/domains-hosts/bing_domain_web
recon/domains-hosts/brute_hosts
recon/domains-hosts/builtwith
recon/domains-hosts/certificate_transparency
recon/domains-hosts/google_site_web
recon/domains-hosts/hackertarget
recon/domains-hosts/mx_spf_ip
recon/domains-hosts/netcraft
recon/domains-hosts/shodan_hostname
recon/domains-hosts/ssl_san
```

```
recon/domains-hosts/threatcrowd
recon/domains-vulnerabilities/ghdb
recon/locations-locations/geocode
recon/locations-locations/reverse_geocode
recon/locations-pushpins/flickr
recon/locations-pushpins/picasa
recon/locations-pushpins/shodan
recon/locations-pushpins/twitter
recon/locations-pushpins/youtube
recon/netblocks-companies/whois_orgs
recon/netblocks-hosts/reverse_resolve
recon/netblocks-hosts/shodan_net
recon/netblocks-hosts/virustotal
recon/netblocks-ports/census_2012
recon/netblocks-ports/censysio
recon/ports-hosts/migrate_ports
recon/profiles-contacts/dev_diver
recon/profiles-contacts/github_users
recon/profiles-profiles/namechk
recon/profiles-profiles/profiler
recon/profiles-profiles/twitter_mentioned
recon/profiles-repositories/github_repos
recon/repositories-profiles/github_commits
recon/repositories-vulnerabilities/gists_search
recon/repositories-vulnerabilities/github_dorks
reporting/csv
```

The number of available modules to choose from can be overwhelming, especially to a new user, so let's start with a basic example of adding a few target domains to Recon-NG using the `add domains` command:

```
[recon-ng] [OSINT] > add domains www.uberpeople.net
[recon-ng] [OSINT] > add domains www.dualxcrypt.com
```

After adding entries to your Recon-NG database, you can quickly see what you have stored in your local database by typing `show` and pressing the Tab key:

```
[recon-ng] [OSINT] > show
banner      dashboard    leaks        options      repositories
companies   domains      locations    ports        schema
contacts    hosts        modules      profiles     vulnerabilities
credentials keys          netblocks    pushpins     workspaces
```

This will give you a list of every type of result you can “show.” Since we have just added some domains, let's show them by typing `show domains`:

```
[recon-ng] [OSINT] > show domains
```

```
+-----+
| rowid |      domain      | module |
+-----+
| 1     | www.uberpeople.net | user_defined |
| 2     | www.dualxcrypt.com | user_defined |
+-----+
```

```
[*] 2 rows returned
```

```
[recon-ng] [OSINT] >
```

Searching for Modules

You have two ways to search for modules. The previously described method of typing **use** and pressing the Tab key will show you all available modules. If you would rather not look through the entire list, you can type **search** and a string. Let's see what is available for resolving domain names:

```
[recon-ng] [OSINT] > search resolve
```

```
[*] Searching for 'resolve'...
```

```
Recon
```

```
-----
```

```
recon/hosts-hosts/resolve
```

```
recon/hosts-hosts/reverse_resolve
```

```
recon/netblocks-hosts/reverse_resolve
```

Before initiating any type of scans on a target, it is good practice to get into the habit of entering as much information as you can about the target. Doing so may turn up interesting results in a different (and unexpected) module.

Since we don't know much about DualXCrypt, we might turn up some interesting results if we also look for the company name in addition to the domain name. To add a company name, we follow the same process using the command **add companies** *companyname* *description*:

```
[recon-ng] [OSINT] > add companies DualXCrypt na
```

Using Modules

Since we know the domain, we can start our investigation by performing a simple resolve search. A resolve search is a short way of saying we will be performing a DNS Resolution search, which will return the domain's nameservers.

Let's start by using the **hosts-hosts/resolve** module:

```
[recon-ng] [OSINT] > use recon/hosts-hosts/resolve
```

Now that we are using the module, we can show its options using the `show options` command:

```
[recon-ng] [OSINT] [resolve] > show options
```

Name	Current Value	Required	Description
-----	-----	-----	-----
SOURCE	default	yes	source of input

This module requires us to set the source of our input (i.e., the target address of the scan). Let's set this option to `uberpeople.net` by using the `set source` command:

```
[recon-ng] [OSINT] [resolve] > set source uberpeople.net
SOURCE => uberpeople.net
```

Now that we've set the required source information, we can kick off the scan by typing `run`.

```
[recon-ng] [OSINT] [resolve] > run

[*] uberpeople.net => 184.154.76.235
```

That was a pretty simple search that gave us the IP address of our source domain. Now let's try running `reverse resolve` by starting with a source IP address. This can be useful if you have an offending IP address and want to see if any domains are associated with that IP:

```
[recon-ng] [OSINT] [resolve] > use recon/hosts-hosts/reverse_resolve
[recon-ng] [OSINT] [reverse_resolve] > set SOURCE 184.154.76.235

SOURCE => 184.154.76.235

[recon-ng] [OSINT] [reverse_resolve] > run

[*] [host] vps.uberpeople.net (184.154.76.235)

-----
SUMMARY
-----
[*] 1 total (1 new) hosts found.
```

Every time we find a new entity within Recon-NG, it will be automatically added to our working database. To verify this, we can see our new host by typing `show hosts`:


```
[recon-ng] [OSINT] [reverse_resolve] > show hosts
```

```
+-----+
| rowid | host | ip | region | country | latitude | longitude | module |
+-----+
| 1 | vps.uberpeople.net | 184.154.76.235 | | | | reverse_resolve |
+-----+
```

```
[*] 1 rows returned
```

The information in our database can now be used to run additional scans. Let's see if we can find any new information on our host using the `virustotal` module.

The `virustotal` module will look for information on any of the domains listed in the `hosts` table, in our case `vps.uberpeople.net`, against VirusTotal .com's database of domain information. If any new information is found, Recon-NG will automatically update our `hosts` table:

```
[recon-ng] [OSINT] > use recon/hosts-hosts/virustotal
```

```
[recon-ng] [OSINT] [virustotal] > run
```

```
-----
184.154.76.235
-----
```

```
[*] [host] bedmaybebreakfast.com (184.154.76.235)
[*] [host] bedmaybebreakfast.uberpeople.net (184.154.76.235)
[*] [host] beepiam.co (184.154.76.235)
[*] [host] livewithanyone.com (184.154.76.235)
[*] [host] mokotoband.uberpeople.net (184.154.76.235)
[*] [host] ns1.uberpeople.net (184.154.76.235)
[*] [host] ns2.uberpeople.net (184.154.76.235)
[*] [host] uberpeople.net (184.154.76.235)
[*] [host] www.bedmaybebreakfast.com (184.154.76.235)
[*] [host] www.bedmaybebreakfast.uberpeople.net (184.154.76.235)
[*] [host] www.mokotoband.uberpeople.net (184.154.76.235)
[*] [host] www.uberpeople.net (184.154.76.235)
```

```
-----
185.165.169.124
-----
```

```
[*] [host] dualxcrypt.com (185.165.169.124)
```

```
-----
SUMMARY
-----
```

```
[*] 13 total (12 new) hosts found.
```

This was a great find—12 new hosts found!

To be thorough, let's also check our results using the `hackertarget` module to see if we can come up with any additional findings:

```
[recon-ng] [OSINT] > use recon/domains-hosts/hackertarget
[recon-ng] [OSINT] [hackertarget] > run
```

```
-----
DUALXCRYPT.COM
-----
[*] [host] dualxcrypt.com (185.165.169.124)
[*] [host] www.dualxcrypt.com (35.228.80.43)
```

```
-----
SUMMARY
-----
[*] 10 total (2 new) hosts found.
```

While this may not appear to be anything useful, this was actually a very important find. Looking closely at the results, we can see that our target host is using different IP addresses for its `www` and non-`www` URLs. This is a good reminder that when entering host information, you should always remember to enter both `www.domain.com` and `domain.com`.

With our new IP information, let's go back and rerun the `virustotal` module to see if our results have changed:

```
[recon-ng] [OSINT] > use recon/hosts-hosts/virustotal
[recon-ng] [OSINT] [virustotal] > run
```

```
-----
185.165.169.124
-----
[*] [host] dualxcrypt.com (185.165.169.124)

-----
35.228.80.43
-----
[*] [host] blog.dualxcrypt.com (35.228.80.43)
[*] [host] blogs.dualxcrypt.com (35.228.80.43)
[*] [host] dualxcrypt.com (35.228.80.43)
[*] [host] liearey1.com (35.228.80.43)
[*] [host] my-vidar.com (35.228.80.43)
[*] [host] new.my-vidar.com (35.228.80.43)
[*] [host] old-vidar.com (35.228.80.43)
[*] [host] resources.old-vidar.com (35.228.80.43)
[*] [host] secretdomain912.com (35.228.80.43)
[*] [host] www.liearey1.com (35.228.80.43)
[*] [host] www.my-vidar.com (35.228.80.43)
[*] [host] www.old-vidar.com (35.228.80.43)
[*] [host] www.secretdomain912.com (35.228.80.43)
```

```

-----
SUMMARY
-----
[*] 26 total (13 new) hosts found.

```

Look at that! We have now discovered quite a few domain names that are associated with our target's IP address, which we will investigate further in the next chapter. For now, let's stick with looking for network-related target information.

Looking for Ports with Shodan

Shodan is a great “ninja” resource because you can use it to find open ports on a target without actually having to touch the target yourself. It is full stealth!

To see which Shodan modules are available, type `search shodan` to return a list of available modules:

```

[recon-ng] [OSINT] [hackertarget] > search shodan
[*] Searching for 'shodan'...

Recon
-----
recon/domains-hosts/shodan_hostname
recon/hosts-ports/shodan_ip
recon/locations-pushpins/shodan
recon/netblocks-hosts/shodan_net

```

Four Shodan modules are available. To search by IP address, we can use the `shodan_ip` module:

```

[recon-ng] [OSINT] > use recon/hosts-ports/shodan_ip
[recon-ng] [OSINT] [shodan_hostname] > run

-----
WWW.UBERPEOPLE.NET
-----
[*] Searching Shodan API for: hostname:www.uberpeople.net

-----
UBERPEOPLE.NET
-----
[*] Searching Shodan API for: hostname:uberpeople.net
[*] [port] 184.154.76.235 (993/<blank>) - vps.uberpeople.net
[*] [host] vps.uberpeople.net (184.154.76.235)

-----
DUALXCRYPT.COM
-----
[*] Searching Shodan API for: hostname:dualxcrypt.com

```

```
-----  
SUMMARY  
-----  
[*] 1 total (0 new) hosts found.  
[*] 1 total (1 new) ports found.
```

Bingo! We've found a new open port.

We can even take this one step further and search a range of IP addresses. Unlike the previous Shodan module that searched for hosts against a single IP address, the `shodan_net` module will search for hosts against an entire netblock (i.e., a range of IP addresses).

Larger organizations will typically own an entire block of IP addresses, so running this type of search is probably more useful if you are performing recon on an organization.

Another example of why it might be useful to search an entire block of IPs is to see what other domains may be hosted with the same ISP. For example, malicious ISPs will typically host more than one malicious site, so there is a good chance that a site hosting stolen credit card numbers will be surrounded by other similar sites. Learning about these sites may provide additional context or clues to your investigation.

To do this, we use the `shodan_net` module, then set the source of the target netblock:

```
[recon-ng] [default] > load recon/netblocks-hosts/shodan_net  
[recon-ng] [default] [shodan_net] > set SOURCE 184.154.76.235/24  
SOURCE => 184.154.76.235  
[recon-ng] [default] [shodan_net] > run
```

As you can probably imagine, scanning the entire /24 netblock returned a lot of of unrelated results. I would not recommend running this type of scan unless you have a good reason to. Remember, anything you discover will be automatically added to your Recon-NG database, so randomly scanning IPs will load you down with worthless results and increase future scan times!

Summary

This chapter covered three major information reconnaissance and discovery tools: SpiderFoot, Intrigue.io, and Recon-NG. The tools are similar in nature, but each contains its own “secret sauce” for how it is able to build on previous results to automate the discovery and collection process. SpiderFoot and Intrigue.io are considered *automated* discovery tools, while Recon-NG is considered a *manual* discovery tool.

The advantage of using a manual approach to information gathering is that you have control over the flow of information being returned to you. Using fully automated tools can be a great way to save time, but they can also leave you with the tedious task of having to review hundreds (or even thousands) of unwanted results. The flip side is that automated tools can also save you a considerable amount of time and deliver unexpected positive results by looking in places you would not have considered.

It all comes down to your own personal preferences and how you prefer working with tools and information.

The next few chapters focus on discovering and collecting information from websites and web applications.

Web Exploration

In This Part

Chapter 6: Website Information Gathering

Chapter 7: Directory Hunting

Chapter 8: Search Engine Dorks

Chapter 9: WHOIS

Chapter 10: Certificate Transparency and Internet Archives

Chapter 11: Iris by DomainTools

With Part I of this book focusing on researching and exploring network addresses, Part II will now focus on gathering and extracting information and intelligence from websites and domains.

This part will cover topics like fingerprinting and gathering baseline information on target websites and move into more complex web scenarios like attempting to brute-force folders on a website in order to find valuable information.

This part will also cover topics like leveraging advanced search techniques (dorking) to discover hidden information on a website and will work the process of domain attribution using tools like WHOIS, certificate transparency logs, the Wayback Machine (archive.org), and the DomainTools IRIS platform.

NOTE I will be completely honest and say that the process of trying to attribute a domain with an owner absolutely sucks. It can be brutal, and all of the new

GDPR privacy regulations are making the process that much more difficult. Even more so if you are hunting threat actors—they will not make the process easy for you. Depending on the domain or the target, attribution can often come down to simply having access to the right historical data—and that information is usually not free (or cheap). When dealing with threat actors, finding a true domain owner will often involve pivoting your way through a maze of fake names, domains, and burner accounts. But rest assured, with access to the right historical data, there will often be repeats in information that you can track. More often than not, the answer is there—you just have to be willing to put in the time to find it.

Website Information Gathering

As we move on to the next step of intelligence gathering, we can now start to look at websites themselves. Fingerprinting a website is the absolute first step in developing a game plan for further investigation. When performing attack surface discovery against a target, or even an entire organization, you can learn a tremendous amount of information by performing a little bit of passive reconnaissance. This is where our journey begins.

WARNING It is important to note that even though many of the tools we will cover contain brute forcing options, those will be beyond the scope of this book. If you are looking for a great web recon tool that also does exploit testing, I recommend you check out Sn1per (<https://github.com/1N3/Sn1per>). We won't be talking about Sn1per because the majority of its uniqueness is in its exploit capabilities, which is outside the scope of this book. Sn1per integrates with Nikto, WPScan, and a number of other tools to perform its tests—it is definitely worth checking out if you need a tool that can also break into a target.

BuiltWith

As a starting point, it is generally a good idea to understand which technologies are in use on your target's web app. Builtwith.com is a great place to start. BuiltWith is designed to be a lead generation and sales tool that allows

salespeople to identify technologies present on a website or web application. As it happens, it's also great for initial recon on a website or web application.

You can run your searches directly from its website, or if you want more detailed information, register for an API key. The free API key allows you 10 free searches per month.

To start, let's look up my personal website, www.vinnytroia.com, shown in Figure 6.1.

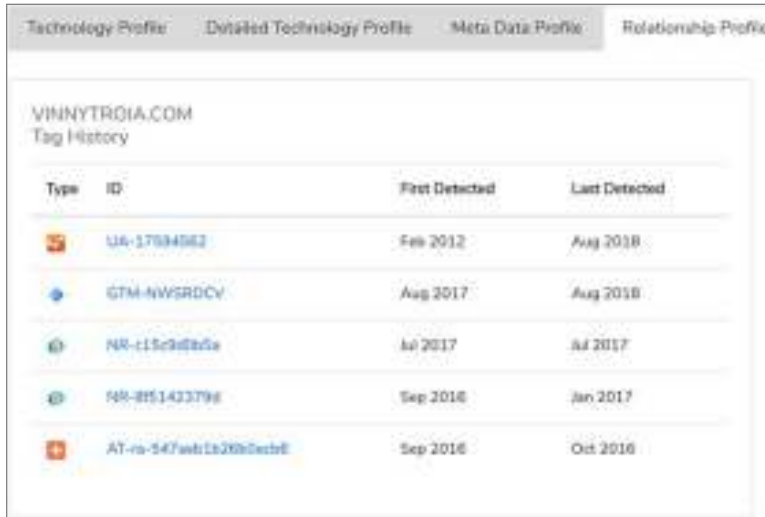
VINNYTROIA.COM			
Technology Profile			
Detailed Technology Profile			
VINNYTROIA.COM			
Analysis and Tracking	First Detected	Last Detected	
Google Analytics Application Performance, Analytics, Measurement, Virtual Event Tracking	Nov 2013	Dec 2018	
Google Universal Analytics	Nov 2018	Dec 2018	
New Relic Application Performance	Jul 2015	Mar 2019	
Google Analytics Classic	Nov 2015	Dec 2017	
Leadin Facebook Forms and Surveys, Marketing Automation	Nov 2015	Mar 2017	
HubSpot Marketing Automation	Nov 2016	Mar 2017	
HubSpot Forms Marketing Automation	Nov 2017	Mar 2017	
MediaKern Screenshot, Tracking, Content and Analytics	Nov 2015	Dec 2015	
Outlogic Advertising Tracking, Conversion, Content and Analytics	Oct 2019	Oct 2019	
Lokaisa Crowd Control	Oct 2018	Oct 2018	
Wiggins			
Yoast SEO Premium WordPress Plugin	Nov 2015	Dec 2018	
Gravity Forms Form and Survey Plugin	Nov 2017	Dec 2018	

Figure 6.1

Looking at the detailed Technology Profile tab, we can get an idea of what software/technology is running—and has historically run—on my site. We can use this to build an attack strategy such as looking for plugin or application vulnerabilities.

Finding Common Sites Using Google Analytics Tracker

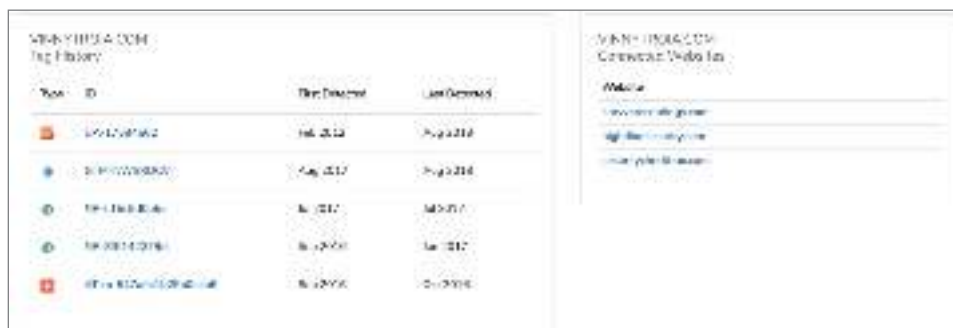
Heading over to the Relationship Profile tab, we can see the Google Analytics ID in use on this site. You can often find related websites or subdomains by looking at the GA tracking ID. More often than not, people will use the same Google account to set up analytics tracking for multiple websites. Analyzing the ID can give you a clue as to some of their other domains. The Relationship Profile tab (Figure 6.2) will show a history of tags used on the site.



Type	ID	First Detected	Last Detected
UA	UA-17894582	Feb 2012	Aug 2018
GTM	GTM-NW5RDCV	Aug 2017	Aug 2018
NR	NR-115c9d8a5a	Jul 2017	Jul 2017
NR	NR-8f5143379e	Sep 2016	Jan 2017
AT	AT-ra-54f7a615a29b0e0c0e	Sep 2016	Oct 2016

Figure 6.2

Looking further into the GA ID for my website, Figure 6.3 shows that the same ID is also in use for three other websites (which also belong to me).



Type	ID	First Detected	Last Detected
UA	UA-17894582	Feb 2012	Aug 2018
GTM	GTM-NW5RDCV	Aug 2017	Aug 2018
NR	NR-115c9d8a5a	Jul 2017	Jul 2017
NR	NR-8f5143379e	Sep 2016	Jan 2017
AT	AT-ra-54f7a615a29b0e0c0e	Sep 2016	Oct 2016

Created Website
vinnytroia.com
vinnytroia.org
vinnytroia.net

Figure 6.3

IP History and Related Sites

Another useful feature of BuiltWith is the ability to see which other domains are currently using—or have historically used—the same IP address. If you are looking for a specific threat actor or organization, it is likely that they will recycle the same IP addresses, especially if they run their own servers. This is a good way to get a glimpse at some of the other domains they have.

Figure 6.4 shows the history of the IP address associated with my website, www.vinnytroia.com.



Figure 6.4

All of these examples were shown using the free version of [BuiltWith.com](#). There is also a premium version geared more toward generating sales leads.

Webapp Information Gatherer (WIG)

WIG is exactly what the name says—it is a tool to gather information on web apps. Since most websites/web apps use content management systems, WIG is a good way to quickly identify which CMS the target is using.

NOTE WIG is available at <https://github.com/jekyc/wig>.

CMS systems power a large majority of modern websites, so being able to identify which CMS is in use at a target website is a fairly important first step in determining the direction for the rest of the engagement.

Parameters for WIG include:

```
-l: list.txt
-a: do not stop after the first CMS is detected
-u: specify a user agent
-N: do not load cached responses
-w: specify an output file
```

To start, we are going to run a simple WIG with no additional parameters against my personal site, vinnytroia.com:

```
python3 wig.py vinnytroia.com
```

TIP When a web browser connects to a website, it includes a `user-agent` field in the HTTP header. This allows the website to identify the browser that is connecting to it. Each browser has its own distinct user agent, and this can often be a way for web application firewalls to block scans from occurring. An important feature of WIG, and of any good scanner, is the ability to mask your user agent. WIG also supports the ability to scan a number of sites at once using the `-l` parameter.

SITE INFO		
IP	Title	
192.241.180.214	Vinny Troia	

VERSION		
Name	Versions	Type
WordPress	5.0.2	CMS
Apache	2.4.10 2.4.11 2.4.12 2.4.5 2.4.6	Platform
nginx	2.4.9	Platform
PHP	7.1.25	Platform

INTERESTING		
URL	Note	Type
/wp-login.php	Wordpress login page	Interesting
/login/	Login Page	Interesting
/readme.html	Readme file	Interesting
/robots.txt	robots.txt index	Interesting
/test/	Test directory	Interesting
/test.php	Test file	Interesting

Time: 9.2 sec	Urls: 748	Fingerprints: 40401
---------------	-----------	---------------------

We can quickly see some of the status of my personal website, which includes my WordPress login page URL and that I have a `/test/` folder and a `test.php` file (which I completely forgot about and will now be removing).

NOTE Until I ran this scan, I had no idea the `test.php` file still existed. I actually don't even remember creating it which is exactly my point. These are *exactly* the types of leftover files that can sink entire ships. I hope this example illustrates the importance for organizations and CISOs to ensure tests are being run against their own websites.

I have personally worked for organizations that had rogue servers lingering on their network. People forget, jobs turn over, administrations change. The larger the organization becomes, the more likely things will be forgotten. And when dealing with servers, if admins do not remember the server exists or know to look for it, there is a good chance that server will be excluded from patch cycles, regular admin password resets, and so on.

But why scan one site when we can scan many sites?

Large organizations may not even realize what web apps are running on their network, so this is a great way to quickly fingerprint everything in a given environment.

For our next example, I've loaded five different sites into a file called `targets.txt`. (For those playing along at home, you can pick any number of sites and put them in a file—one site per line). We will use WIG to scan all of them, while changing our user agent and outputting all results to a file called `results.txt`.

The following code will tell `wig.py` to scan all targets in `target.txt`, tell it to use a specific browser user agent when performing the scans, and output everything to `results.txt`:

```
python3 wig.py -l targets.txt -u "Mozilla/5.0 (Android 4.4; Mobile; \
rv:41.0) Gecko/41.0 Firefox/41.0" -N -w results.txt
```

The output of the file is JSON, so if you don't capture the initial output text, you will have to parse the JSON into Excel or whatever tool you are using to read the results:

```
wig - WebApp Information Gatherer
```

```
Scanning https://www.pepsi.com...
```

SITE INFO	
IP	Title
23.63.197.180	Pepsi.com

VERSION		
Name	Versions	Type
Apache		Platform

SUBDOMAINS		
Name	Page Title	IP

https://cms.pepsi.com:443 Pepsi.com 18.222.154.81

Time: 29.0 sec Urls: 72 Fingerprints: 40401
Scanning https://www.nightlionsecurity.com...

SITE INFO	
IP	Title
192.241.180.214	Cyber Security Firm: Penetration Testing, Risk Assessments

VERSION		
Name	Versions	Type
WordPress	4.1.2 4.1.3 4.1.4 4.1.5 4.2 4.2.1 4.2.2	
nginx		Platform
PHP	7.1.25	Platform

INTERESTING		
URL	Note	Type
/readme.html	Readme file	Interesting

VULNERABILITIES		
Affected	#Vulns	Link
WordPress 4.2.1	1	http://cvedetails.com/version/184019
WordPress 4.2.2	2	http://cvedetails.com/version/185073

Time: 16.3 sec Urls: 240
Fingerprints: 40401

Scanning https://www.comodo.com...

SITE INFO	
IP	Title
104.16.21.160	Comodo Global Leader in Cyber Security
104.16.20.160	
104.16.18.160	
104.16.22.160	
104.16.19.160	

VERSION		
Name	Versions	Type
cloudflare		Platform

SUBDOMAINS		
Name	Page Title	IP
http://m.comodo.com:80	Mobile Antivirus Comodo	91.199.212.187
https://m.comodo.com:443	Mobile Antivirus	91.199.212.187
http://blog.comodo.com:80	Comodo News and Internet ...	178.255.86.141

https://blog.comodo.com:443 Comodo News and Internet ... 178.255.86.141

INTERESTING		
URL	Note	Type
/robots.txt	robots.txt index	Interesting
/login/	Login Page	Interesting

Time: 49.1 sec Urls: 716 Fingerprints: 40401

Scanning http://whitepacket.com...

SITE INFO	
IP	Title
104.24.119.111	WhitePacket Home
104.24.118.111	

VERSION	
Name	Versions
WordPress	4.8.8
cloudflare	
Apache	2.2.11 2.2.12 2.2.13 2.2.14 2.2.15 2.2.18 2.2.19 2.2.20 2.2.21 2.2.22 2.2.23 2.2.25 2.2.26 2.2.27 2.2.28 2.2.29 2.3.0 2.3.10 2.3.11 2.3.12 2.3.13 2.3.14 2.3.15 2.3.2 2.3.3 2.3.4 2.3.5 2.3.6 2.3.7 2.3.8 2.3.9 2.4.0 2.4.1 2.4.2 2.4.3
PHP	5.4.45-0+deb7u12

INTERESTING		
URL	Note	Type
/robots.txt	robots.txt index	Interesting
/readme.html	Readme file	Interesting
/login/	Login Page	Interesting

TOOLS		
Name	Link	Software
wpscan	https://github.com/wpscanteam/wpscan	WordPress
CMSmap	https://github.com/Dionach/CMSmap	WordPress

Time: 189.7 sec Urls: 410 Fingerprints: 40401

Now that we have the fundamentals of WIG down, let's move on to a more advanced CMP mapping and discovery tool.

CMSMap

CMSMap is an open-source Python scanner that automates the process of detecting security flaws in popular content management systems (CMSs). CMSMap is similar to WIG in that both tools are able to map and identify CMS systems, but CMSMap is more advanced and comes with additional features.

NOTE You can download CMSMap here: <https://github.com/Dionach/CMSmap>.

Though similar, CMSMap is considerably more advanced than WIG and supports a number of different options and parameters. In addition to just detecting CMSs in use by web apps, CMSMap can also enumerate site plugins, bruteforce logins, and crack password hashes. CMSMap also uses the official ExploitDB repo, which you can download here: <https://github.com/offensive-security/exploitdb.git>.

Some of the available CMSMap parameters include:

```
-f W/J/D/M, --force W/J/D/M
                        force scan (W)ordpress, (J)oomla or (D)upal or (M)oodle
-F, --fullscan         full scan using large plugin lists.
-t, --threads          number of threads (Default 5)
-a, --agent            set custom user-agent
-H, --header           add custom header (e.g. 'Authorization: Basic ')
-i, --input            scan multiple targets listed in a given file
-o, --output           save output in a file
-E, --noedb            enumerate plugins without searching exploits
-c, --nocleanurls      disable clean urls for Drupal only
-s, --nossllcheck      don't validate the server's certificate
-d, --dictattack       run low intense dictionary attack during scan

Brute-Force:
-u, --usr              username or username file
-p, --psw             password or password file
-x, --noxmlrpc        brute-forcing WordPress without XML-RPC

Post Exploitation:
-k, --crack            password hashes file
(Require hashcat installed. For WordPress and Joomla only)
-w, --wordlist         wordlist file

Others:
-v, --verbose         verbose mode (Default false)
-h, --help            show this help message and exit
-D, --default         run CMSmap with default options
-U, --update          use (C)MSmap, (P)lugins or (PC) for both
```


Running a Single Site Scan

Let's start with a basic scan of a website running the Drupal CMS software. To find some sites running Drupal, we can head over to www.builtwith.com, and search for "Drupal."

For a test scan, I randomly selected PayChex.com, a popular payroll company. There is a good chance it has a web application firewall (WAF), and most WAFs know to block for CMS scanning apps. The way to get around that is to specify a custom browser user agent, which will trick the WAF into thinking the request is coming from a user's web browser.

The `-a` parameter of CMSMap will allow us to specify a custom user agent:

```
root@OSINT:/opt/CMSmap: python3 cmsmap.py -s -a \
"Mozilla/5.0 (Android 4.4; Mobile; rv:41.0) Gecko/41.0 Firefox/41.0" \
https://www.paychex.com
[-] Date & Time: 01/01/2019 15:48:12
[I] Threads: 5
[-] Target: https://www.paychex.com (104.17.169.11)
[I] Server: cloudflare
[L] X-Generator: Drupal 8 (https://www.drupal.org)
[L] X-Frame-Options: Not Enforced
[L] Robots.txt Found: https://www.paychex.com/robots.txt
[I] CMS Detection: Drupal
[I] Drupal Theme: custom
[M] EDB-ID: 29019 "Zikula CMS 1.3.5 - Multiple Vulnerabilities"
[M] EDB-ID: 41564 "Drupal 7.x Module Services - Remote Code Execution"
[-] Enumerating Drupal Usernames via "Views" Module...
[-] Enumerating Drupal Usernames via "/user/..."
[-] Drupal Default Files:
[-] Drupal is likely to have a large number of default files
[-] Would you like to list them all? Y

[results truncated]

[-] Search Drupal Modules ...
[I] content
[I] Checking for Directory Listing Enabled ...

[-] Completed in: 0:01:22
```

Scanning Multiple Sites in Batch Mode

Similar to what we did while using WIG, CMSMap will scan multiple sites included in a target file. To do this, use the `-i` switch followed by the name of the file with the target domains. The following example will use `targets.txt`

as our input filename, specify an output file of `output.txt`, and use a custom user agent:

```
root@OSINT:/opt/CMSmap: python3 cmsmap -i targets.txt -o output.txt \
-a "Mozilla/5.0 (Android 4.4; Mobile; rv:41.0) Gecko/41.0 Firefox/41.0"
```

An annoying part about scanning sites in batch is that if a CMS is not detected with one of the sites, the script won't automatically move to the next site. So you have to either force the CMS to a known type or just scan the sites one at a time. The following output shows what happens if site detection fails:

```
[~] Date & Time: 01/01/2019 20:45:12
[L] Robots.txt Found: http://www.nightlionsecurity.com/robots.txt
[ERROR] CMS detection failed :(
[ERROR] Use -f to force CMSmap to scan (W)ordpress, (J)oomla or (D)rupal
```

Detecting Vulnerabilities

An amazing feature of CMSMap is its ability to detect vulnerabilities within target sites. The following output was taken from a scan run with the exact parameters shown in the preceding code. We have to leave out the names of the websites because of the discovered vulnerabilities, but you can quickly see how powerful this tool is:

```
[I] Threads: 5
[~] Target: http://www.fakebank.com (52.37.170.23)
[M] Website Not in HTTPS: http://www.fakebank.com
[I] Server: Apache/2.4.10 (Debian)
[L] X-Frame-Options: Not Enforced
[I] Strict-Transport-Security: Not Enforced
[I] X-Content-Security-Policy: Not Enforced
[I] X-Content-Type-Options: Not Enforced
[L] Robots.txt Found: http://www.fakebank.com/robots.txt
[I] CMS Detection: WordPress
[I] Wordpress Version: 4.8.8

[M] EDB-ID: 44949 "WordPress Core < 4.9.6 - (Authenticated)
Arbitrary File Deletion"
[I] Wordpress Theme: Avada
[M] EDB-ID: 34511 "Multiple WordPress Themes -
'admin-ajax.php?img' Arbitrary File Download"
[~] WordPress usernames identified:
[M] [omitted]
[M] [omitted]
[M] [omitted]
```

```
[M] [omitted]
[M] [omitted]
[M] [omitted]
[M] [omitted]
[M] [omitted]
[M] [omitted]

[M] XML-RPC services are enabled
[M] Website vulnerable to XML-RPC Brute-Force Vulnerability
[I] Autocomplete Off Not Found: http://www.fakebank.com/wp-login.php

[-] Default WordPress Files:
[I] http://www.fakebank.com/license.txt
[I] http://www.fakebank.com/readme.html
[-] Searching Wordpress Plugins ...
[I] google-analytics-for-wordpress v6.2.6
[I] revslider
[I] akismet

[M] EDB-ID: 37826 "WordPress 3.4.2 -
Multiple Path Disclosure Vulnerabilities"
[M] EDB-ID: 37902 "WordPress Plugin Akismet -
Multiple Cross-Site Scripting Vulnerabilities"
[I] fusion-core
[I] fusion-builder
[I] feed

[M] EDB-ID: 38624 "WordPress Plugin WP Feed - 'nid' SQL Injection"
[I] Checking for Directory Listing Enabled ...
Checking for Directory Listing Enabled ...
[-] Date & Time: 01/01/2019 21:00:26
[-] Completed in: 0:04:58
```

In addition to detecting vulnerabilities, CMSMap can enumerate which themes or plugins are in use on a site, and even bruteforce CMS systems using a given username and password file. It is an incredibly powerful middle-tier tool—it has just enough features without feeling overly complex and advanced.

Now let's move on to our third and final tool, WPScan.

WPScan

Wpscan is a black box information gathering tool and vulnerability scanner for Wordpress. In my opinion, it is by far the best on the market and has become the 'NMAP' of vulnerability scanning.

NOTE WPScan is available at <https://wpscan.org>.

Right now you may be thinking: CMSMap already scans WordPress. Why do I need another scanner? The answer is it depends on how deep and thorough you want to get. CMSMap is focused on providing similar functionality to multiple types of CMS systems, whereas WPScan provides advanced features solely focused on WordPress sites.

Other tools may provide a good baseline for scanning and searching CMS systems, which may be 80% of what you need in the end. If your focus is on scanning or testing WordPress sites, WPScan is it.

The number of parameters available in WPScan can be daunting, but the real value in using this tool is that you can use it to stay completely under the radar and not be detected. Avoiding detection can be tricky, especially when facing off against modern WAFs. WPScan allows you to use proxies and has a built-in stealth mode that will rotate user agents on every request.

Here are some of the more important parameters that we will be discussing:

```
-v, --verbose                Verbose mode
-o, --output FILE            Output to FILE
-f, --format FORMAT          Output results in specified format

--detection-mode MODE        Default: mixed
Available choices: mixed, passive, aggressive

--user-agent, --ua VALUE
--random-user-agent, --rua    Use a random user-agent for each scan
--http-auth login:password
-t, --max-threads VALUE      Max threads to use
--throttle MilliSeconds      MS to wait between each web request
    --request-timeout SECONDS The request timeout in seconds
    --connect-timeout SECONDS The connection timeout in seconds
    --disable-tls-checks      Disables SSL/TLS verification
    --proxy protocol://IP:port
    --proxy-auth login:password
    --cookie-string COOKIE    Cookie string to use in requests
    --cookie-jar FILE-PATH     File to read and write cookies
                                Default: /tmp/wpscan/cookie_jar.txt
--force                      Assume WordPress is running

    --wp-content-dir          Manually set the wp-contents directory
    --wp-plugins-dir

-e, --enumerate [OPTS]      Enumeration Process - includes
ability to enumerate vulnerable plugins, all plugins, themes,
Timthumbs, config backups, database exports, user ids, media ids,
and all

-P, --passwords FILE-PATH    Passwords to use during attack
-U, --usernames LIST          Usernames to use during
```

```
password attack
    --multicall-max-passwords      Maximum number of passwords to
send by request with XMLRPC multicall
    --password-attack ATTACK      Force the supplied attack to be
used rather than automatically determining one
    --stealthy                     Force stealth/passive mode
```

Out of the box, WPScan will automatically enumerate all running plugins on the site, check for vulnerabilities of those plugins, and look for important files like config backups.

After running the tool with no parameters other than a URL against my personal site, we can already see a considerable difference in the output generated between WPScan and other plugins. The following is a condensed version of the results:

```
root@OSINT:/opt/wpscan: wpscan --url http://www.vinnytroia.com
```

WordPress Security Scanner by the WPScan Team

Version 3.4.2

Sponsored by Sucuri - <https://sucuri.net>

@ WPScan , @ethicalhack3r, @erwan lr, @ FireFart

```
[+] URL: http://www.vinnytroia.com/
[+] Started: Thu Jan  3 07:55:46 2019
```

Interesting Finding(s) :

```
[+] http://www.vinnytroia.com/
| Interesting Entries:
|   - Server: nginx
|   - X-Powered-By: PHP/7.1.25, PleskLin
|   - Access-Control-Allow-Origin: *
|   - Access-Control-Allow-Credentials: true
|   - Access-Control-Allow-Headers: Content-Type,Accept
|   - Access-Control-Allow-Methods: GET, POST, OPTIONS, PUT, DELETE
| Found By: Headers (Passive Detection)
| Confidence: 100%
```

```
[+] http://www.vinnytroia.com/robots.txt
| Found By: Robots Txt (Aggressive Detection)
```

```

| Confidence: 100%

[+] http://www.vinnytroia.com/xmlrpc.php
| Found By: Direct Access (Aggressive Detection)
| Confidence: 100%

[+] http://www.vinnytroia.com/readme.html
| Found By: Direct Access (Aggressive Detection)
| Confidence: 100%

[+] WordPress version 5.0.2 identified (Latest, released on 2018-12-19).
| Detected By: Rss Generator (Passive Detection)
| - http://www.vinnytroia.com/feed/,

[+] WordPress theme in use: jupiter-child
| Location: http://www.vinnytroia.com/wp-content/themes/jupiter-child/
| Style URL: http://www.vinnytroia.com/wp-content/themes/
jupiter-child/style.css?ver=5.0.2
| Style Name: Jupiter Child Theme
| Style URI: http://themeforest.net/user/artbees
| Description: Child theme for the Jupiter theme...
| Author: Your name here

[+] Enumerating All Plugins
[+] Checking Plugin Versions

[i] Plugin(s) Identified:

[+] contact-form-7

[+] google-analytics-for-wordpress

[+] google-analytics-premium

[+] gravityforms
| Location: http://www.vinnytroia.com/wp-content/plugins/gravityforms/
| Detected By: Urls In Homepage (Passive Detection)
| Version: 2.4.4 (100% confidence)

[+] js_composer

[+] js_composer_theme
| Location: http://www.vinnytroia.com/wp-content/plugins/...

[+] kiwi-logo-carousel
| Location: http://www.vinnytroia.com/wp-content/plugins/...

[+] masterslider
| Location: http://www.vinnytroia.com/wp-content/plugins/masterslider/

```

```
[+] rdv-youtube-playlist-video-player
| Location: http://www.vinnytroia.com/wp-content/plugins/...

[+] tubePress_pro_5_1_5
| Location: http://www.vinnytroia.com/wp-content/plugins/...
|
| Detected By: Urls In Homepage (Passive Detection)
|
| The version could not be determined.

[+] wordpress-seo
| Location: http://www.vinnytroia.com/wp-content/plugins/wordpress-seo/
| Latest Version: 9.3 (up to date)
| Last Updated: 2018-12-18T09:25:00.000Z

[+] wp-super-cache
| Location: http://www.vinnytroia.com/wp-content/plugins/...
| Latest Version: 1.6.4 (up to date)
| Last Updated: 2018-12-20T09:36:00.000Z

[+] Enumerating Config Backups
Checking Config Backups - Time: 00:00:03

[i] No Config Backups Found.

[+] Finished: Thu Jan 3 07:55:59 2019
[+] Requests Done: 92
[+] Cached Requests: 5
[+] Data Sent: 18.989 KB
[+] Data Received: 2.482 MB
[+] Memory used: 67.148 MB
[+] Elapsed time: 00:00:13
```

The results of my site are pretty boring. There were no vulnerabilities detected and no real action that can be taken from these results outside of trying to bruteforce my login/password. From an OSINT perspective, there is nothing interesting from these results (to my credit, I keep the site updated and the server patched specifically for this reason).

Now let's try something a little harder.

Dealing with WAFs/WordPress Not Detected

If we run the same scan against my company website, www.NightLionSecurity.com, we get a different message:

```
root@OSINT:/opt/wpscan: wpscan --url http://www.nightlionsecurity.com

Scan Aborted: The remote website is up, but does not seem
to be running WordPress.
```

More often than not, you can tell a site is running WordPress by just going to the `/wp-admin/` folder—for example, `http://www.nightlionsecurity.com/wp-admin`.

If you are sure the site is running WordPress and you are still getting this message, then the site has a WAF in place that needs to be bypassed. This happens to be the case for NightLion's website. There are a few things we can do to work around this error message.

The first thing we can try is to modify our parameters to use a random user agent. We can also use the `--force` switch because we can manually verify that WordPress is running on the domain:

```
root@OSINT:/opt/wpscan: wpscan --url http://www.nightlionsecurity.com \
--random-user-agent --force
```

```
[+] URL: http://www.nightlionsecurity.com/
[+] Effective URL: https://www.nightlionsecurity.com/
[+] Started: Thu Jan 3 02:35:16 2019
```

Interesting Finding(s):

```
[+] https://www.nightlionsecurity.com/
| Interesting Entries:
|   - Server: nginx
|   - X-Powered-By: PHP/7.1.25, PleskLin
|   - Access-Control-Allow-Origin: cdn.nightlionsecurity.com
| Found By: Headers (Passive Detection)
| Confidence: 100%
```

```
[i] The WordPress version could not be detected.
```

While these results aren't great, they are definitely a step in the right direction. In an ideal situation, we would see results that include things like username and plugin enumeration, checks for leftover files, and site vulnerabilities.

Taking this a step further, let's see if we can find any vulnerable plugins or usernames. At this point I would typically use a combination of parameters to try to get results.

The `--stealthy` parameter combines a few different parameters to create the least noisy scan possible. Stealthy mode includes a random user agent, and also takes a less aggressive approach to plugin detection by increasing the time between probes.

The next step would be to use the `--proxy` switch to send every request through a proxy (that will use a different IP address for every request).

Finally, we can ensure that site usernames are enumerated by using `--enumerate u`. Typically this is included in a default scan but is skipped when scanning in stealthy mode.

WARNING The ability to rotate IP addresses does not come with WPScan. Please do not expect this functionality out of the box with this (or any other) tool. I use a *paid* proxy service called Storm Proxies that automatically rotates my IPs on every request. The flow works like this:

- WPScan allows me to use an IP address →
- Storm Proxies provides me with an IP address that I use as my proxy address →
- For each request made to that IP, Storm Proxies relays it through a different address.

Putting it all together, our request looks like this:

```
root@OSINT:/opt/wpscan: wpscan --url http://www.nightlionsecurity.com \
--stealthy --force -proxy 'socks5://127.0.0.1:9050' --enumerate u
```

The new scan resulted in the following output:

```
[+] URL: https://www.nightlionsecurity.com/
[+] Started: Thu Jan 3 04:30:13 2019
```

Interesting Finding(s):

```
[+] https://www.nightlionsecurity.com/
| Interesting Entries:
|   - Server: nginx
|   - X-Powered-By: PHP/7.1.25, PleskLin
|   - Access-Control-Allow-Origin: cdn.nightlionsecurity.com
| Found By: Headers (Passive Detection)
| Confidence: 100%
```

[i] User(s) Identified:

```
[+] Vinny
| Detected By: Rss Generator (Passive Detection)
| Confirmed By: Rss Generator (Aggressive Detection)
```

```
[+] BlogAdmin
```

```
| Detected By: Rss Generator (Passive Detection)
| Confirmed By: Rss Generator (Aggressive Detection)

[+] Editor
| Detected By: Rss Generator (Passive Detection)
| Confirmed By: Rss Generator (Aggressive Detection)

[+] Requests Done: 16
[+] Cached Requests: 63
[+] Data Sent: 3.992 KB
[+] Data Received: 1.437 MB

[i] Config Backup(s) Identified:

[+] http://www.nightlionsecurity.com.com/wp-config.bak
| Detected By: Direct Access (Aggressive Detection)
```

These results are much better—we are clearly making progress. We were now able to use different parameters to get a list of the site usernames (which we were unable to do in the first few examples).

We also found a configuration backup, which is a huge find! *A wp-config backup file should always contain the database username and password, so if you find a backup file, consider it a jackpot.* Admins have been known to leave backups laying around, so definitely check for this.

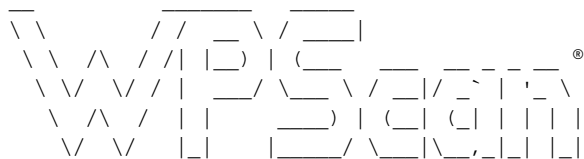
NOTE I added the backup file specifically so this scan would detect it. It is not there anymore (and never really was).

The next step would typically be to identify and look for vulnerabilities in the WordPress plugins. Following that, I would then attempt to bruteforce the login using a common password list.

To look for vulnerable plugins, we can reissue the same command as before but without the `--stealthy` parameter (for this example, using the rotating proxy service is what allows us to fool my WAF). To enumerate the plugins we can use the `--enumerate ap`, or `--enumerate all` to enumerate plugins, as well as usernames, themes, database backups, and everything else.

I don't have many plugins running on my site, so for this final demonstration, let's run our scan against `ManageWP.com` (a WordPress management system for admins):

```
root@OSINT:/opt/wpscan: wpscan --url http://www.managewp.com
--force --random-user-agent --enumerate all
```



WordPress Security Scanner by the WPScan Team

Version 3.4.2

Sponsored by Sucuri - <https://sucuri.net>

@_WPScan_, @ethicalhack3r, @erwan_lr, @_FireFart_

[+] URL: <https://managewp.com/>

[+] Started: Thu Jan 3 04:45:21 2019

Interesting Finding(s):

[+] <https://managewp.com/>

| Interesting Entry: Server: nginx

| Found By: Headers (Passive Detection)

| Confidence: 100%

[+] <https://managewp.com/robots.txt>

| Found By: Robots Txt (Aggressive Detection)

| Confidence: 100%

[+] WordPress version 4.9.8 identified (Insecure, released on 2018-08-02)

| [!] 7 vulnerabilities identified:

|

| [!] Title: [Excluded from Publishing]

| Fixed in: 5.0.1

| References:

| [!] Title: [Excluded from Publishing]

| Fixed in: 5.0.1

| References:

[Results Truncated]

[i] Plugin(s) Identified:

[+] contact-form-7

|

| Detected By: Urls In Homepage (Passive Detection)

| [!] X vulnerability identified:

[+] mailchimp-for-wp

```
[+] wordpress-seo
[!] The version is out of date, the latest version is 9.3
| Detected By: Comment (Passive Detection)
|
| [!] X vulnerability identified:

[Results Truncated]

[i] No Config Backups Found.
```

The results are immediately apparent: We have identified multiple vulnerable WordPress plugins. Now that we have identified vulnerabilities on the target site, the next step will depend on the scope of your engagement.

NOTE The actual list of vulnerabilities was removed for legal reasons.

If you are authorized to try to break into the site, you can look on [exploitdb](#) to try to find a working exploit, or try tools like [Sn1per](#) (discussed at the beginning of this chapter). In many cases, simply reporting the details to the site owner will be enough, so please make sure you have appropriate permissions before actually crossing the line of gathering information and breaking into something.

Summary

This chapter covered some of the tools that can be used to fingerprint a target website and identify which technologies are in use. Since the majority of all modern websites run on a CMS (content management system) application, being able to find information on the type of CMS in use and the existence of any lingering files or vulnerabilities on the platform is crucial. Lingering files can lead to the discovery of major information spillage or even attribution of a threat actor.

Now that we understand how to properly identify the technologies in use on our targets, the next chapter will focus on the art of directory treasure hunting.

Directory Hunting

Inevitably on your quest to find information about a website, you will need to look for hidden (i.e., forgotten) treasures. The best way to do this is to look for random directories. If you have not tried this yet, you will be amazed at how much information people just forget about: webshells, PHPMyAdmin pages, folders with directory browsing and/or full read/write permissions, private files, and so much more. There are several ways to accomplish this. One is bruteforcing the target to find working directories, and the other is analyzing crawl data. This chapter will look at both.

Dirhunt

Dirhunt is a type of web crawler designed to search and analyze directories and folders within a web application. Dirhunt is not really a scraper, and it does not use bruteforce to find folders. Instead, Dirhunt checks a number of sources to find interesting files or folders, including Google and VirusTotal. Dirhunt is also designed to detect false 404 errors to minimize the number of false positives in your results.

NOTE You can download Dirhunt at <https://github.com/Nekmo/dirhunt>.

The parameters of Dirhunt include:

-t, --threads INTEGER	Number of threads to use
-x, --exclude-flags TEXT	Exclude results with these flags
-i, --include-flags TEXT	Only include results with these flags
-e, --interesting-extensions TEXT	Look for files with the following extensions
-f, --interesting-files TEXT	The files with these names are interesting
--stdout-flags TEXT	Return only in stdout the urls of these
--progress-enabled / --progress-disabled	
--timeout INTEGER	
--max-depth INTEGER	Maximum links to follow
--not-follow-subdomains	Subdomains will be ignored
--exclude-sources TEXT	Exclude source engines: robots, virustotal, google
-p, --proxies TEXT	Set one or more proxies to alternate
-d, --delay FLOAT	Delay between requests
--not-allow-redirects	Redirectors will not be followed
--limit INTEGER	Max number of pages processed to search

Let's test Dirhunt's capabilities with no parameters against the Florida Alcohol and Drug Abuse Association (FADAA) website:

```
root@INTEL:/opt/dirhunt: dirhunt https://www.fadaa.org/

[301] https://fadaa.org/ (Redirect)
      Redirect to: https://www.fadaa.org/?
[403] http://fadaa.org/global_inc/ (Generic)
      Index file found: index.php
[301] http://fadaa.org/ (Redirect)
      Redirect to: https://www.fadaa.org/?
[301] http://fadaa.org/global_engine/ (Redirect)
      Redirect to: https://www.fadaa.org/global_engine/Default.asp?
[301] http://www.fadaa.org/global_inc/%2A.css (Redirect)
      Redirect to: https://www.fadaa.org/404.aspx?404;
      http://www.fadaa.org:80/global_inc/*.css
[301] http://www.fadaa.org/global_inc/%2A.js (Redirect)
      Redirect to: https://www.fadaa.org/404.aspx?404;
      http://www.fadaa.org:80/global_inc/*.js
[301] http://www.fadaa.org/ (Redirect)
      Redirect to: https://www.fadaa.org/default.aspx
[403] http://www.fadaa.org/global_inc/ (Generic)
      Index file found: index.php
[200] https://www.fadaa.org/ (Generic)
[404] https://www.fadaa.org/404.aspx (Not Found)
[200] https://www.fadaa.org/default.aspx (Generic)
[302] https://www.fadaa.org/global_engine/Default.asp (Redirect)
```

```

Redirect to: https://www.fadaa.org/global_engine/Default.asp
[302] http://www.fadaa.org/global_engine/ (Redirect)
Redirect to: http://www.fadaa.org/global_engine/
[302] https://www.fadaa.org/global_engine/ (Redirect)
Redirect to: https://www.fadaa.org/global_engine/
[302] https://www.fadaa.org/page/SAMHSA_Treatment (Redirect)
Redirect to: https://findtreatment.samhsa.gov/
[200] https://www.fadaa.org/staff/ (Generic)
Index file found: index.php
[200] https://www.fadaa.org/news/ (Generic)
Index file found: index.php
[404] https://www.fadaa.org/page/ (Not Found)
Index file found: index.php
[200] https://www.fadaa.org/page/resource_links (Generic)
Index file found: index.php
[200] https://www.fadaa.org/networking/ (Generic)
[200] https://www.fadaa.org/page/AOE2017 (Generic)
Index file found: index.php
[200] https://www.fadaa.org/page/Membership (Generic)
Index file found: index.php
[200] https://www.fadaa.org/search/ (Generic)
[302] https://www.fadaa.org/general/ (Redirect)
Redirect to: https://www.fadaa.org/general/
[200] https://www.fadaa.org/login.aspx (Generic)
[302] https://www.fadaa.org/general/register_start.asp (Redirect)
Redirect to: https://www.fadaa.org/general/register_start.asp
[404] https://www.fadaa.org/graphics/ (Not Found) (FAKE 404)
Index file found: index.php
[403] https://www.fadaa.org/global_inc/ (Generic)
[403] https://www.fadaa.org/global_inc/site_templates/js/ (Generic)
[403] https://www.training.fadaa.org/css/ (Generic)
[200] https://www.fadaa.org/page/Healthcare_Division (Generic)
Index file found: index.php
[200] https://www.fadaa.org/page/BusinessDivision (Generic)
Index file found: index.php
[403] https://www.fadaa.org/global_inc/site_templates/ (Generic)
[200] https://www.fadaa.org/page/Boards (Generic)
Index file found: index.php
[302] https://www.fadaa.org/page/Become_a_Member (Redirect)
Redirect to: http://fadaa.site-ym.com/?page=Login
[200] https://www.fadaa.org/page/Housing_Recovery (Generic)
Index file found: index.php
@ Finished after 11 seconds
No interesting files detected "\_("\)/

```

Depending on your results, with this type of scan you can now pivot your strategy to focus on your discoveries. It is extremely likely that you will discover a content management system (CMS) because of how common they are in powering the majority of today's websites. The discovery of a CMS will inev-

itably lead to the discovery of vulnerabilities within that system, which could translate to a quick win for your engagement.

The parameters of Dirhunt can be further tweaked to focus (or not focus) on specific files, file types, subdomains, and more. Looking for compressed file extensions (like .zip, .rar, or .gz) by using the -e switch as shown in the following command can often give you access to lingering files that should have been deleted:

```
dirhunt http://domain.com -e php,zip,sh
```

There is also a good chance you will discover interesting files like leftover configuration files, backups, and log files that may have been forgotten. To specifically look for files, use the -f switch, like this:

```
dirhunt http://domain.com -f access_log,error_log
```

When looking for files, you can also specify a list of files to search from a dictionary file. To specify a dictionary file, use the following:

```
dirhunt http://domain.com -f /var/files/dictfile.txt
```

Wfuzz

Wfuzz has been around for a very long time, and is probably one of the better well-known web application testing tools available. Wfuzz is, in essence, a bruteforcing tool, and probably one of the best available for bruteforcing web applications.

Fuzzing is the process of using automated tools to look for software bugs by sending malformed data to the application. In this case, we are not sending Wfuzz (web fuzz) malformed data. We are using it to send legitimate words and phrases to a web server to check for the existence of hidden or private files and folders. Wfuzz is available for download at <https://github.com/xmendez/wfuzz>.

Parameters for Wfuzz include:

```
Options:
  -h                : This help
  --help            : Advanced help
  --version         : Wfuzz version details
  -e <type>        : encoders/payloads/iterators/
                   : printers/scripts
```


-c	: Output with colors
-v	: Verbose information.
--interact	: This allows you to interact with the program.
-p addr	: Use Proxy in format ip:port:type. Where type could be SOCKS4,SOCKS5 or HTTP if omitted.
-t N	: Specify number of concurrent connections (10 default)
-s N	: Specify time delay between requests (0 default)
-R depth	: Recursive path discovery depth
-L, --follow	: Follow HTTP redirections
-u url	: Specify a URL for the request.
-z payload	: Specify a payload - type,parameters, encoder.
-w wordlist	: Specify a wordlist file (alias for -z file,wordlist).
-V alltype	: All parameters bruteforcing (allvars and allpost).
-X method	: Specify an HTTP method for the request
-b cookie	: Specify a cookie for the requests
-d postdata	: Use post data (ex: "id=FUZZ&catalogue=1")
-H header	: Use header (ex:"Cookie:id=1312321&user=FUZZ")
--basic/ntlm/digest auth	: in format "user:pass" or "FUZZ:FUZZ" or "domain\FUZZ2Z:FUZZ"

Wfuzz contains an almost infinite number of combinations that you can use to “fuzz” a web application. We can do the same when looking for web directories on a target domain.

Wfuzz works by taking your target URL and wordlist and attempts to insert every word in your wordlist wherever and whenever you specify “fuzz” in your URL.

Let’s start with a simple example of trying to bruteforce folders off the main domain path using a wordlist. For our sample wordlist, we are going to use the `big.txt` wordlist in the SecLists repository discussed in Chapter 2. We are also going to use the `--hc 404` tag to hide all 404 responses:

```
root@OSINT:wfuzz -c -z file,/opt/SecLists/Discovery/Web-Content/
big.txt --hc 404 http://www.biz-up.at/FUZZ
```

NOTE If a website has automatic URL redirection set up, you will be flooded with a list of URLs and 302 response codes. This is Wfuzz thinking that every folder is active. If you start to see a bunch of 301 or 302 response codes, then the site may have a custom 404 page, or error redirection set up on every page. In either case, to hide anything but a positive 202 response, use the following code: `--hc 404,301,302`.

Your output will look something like this:

```
=====
ID      Response   Lines      Word        Chars        Payload
=====
000010:  C=403          9 L          24 W          216 Ch      ".bashrc"
000015:  C=403          9 L          24 W          218 Ch      ".htaccess"
000016:  C=403          9 L          24 W          218 Ch      ".htpasswd"
000026:  C=200         551 L        3234 W        66939 Ch     "0"
000919:  C=403          9 L          24 W          218 Ch      "ChangeLog"
000965:  C=403          9 L          24 W          216 Ch      "LICENSE"
000974:  C=404         352 L        2504 W        53514 Ch     "MANIFEST.MF"
001012:  C=403          9 L          24 W          215 Ch      "README"
001015:  C=403          9 L          24 W          215 Ch      "Readme"
001053:  C=403          9 L          24 W          213 Ch      "TODO"
001058:  C=403          9 L          24 W          218 Ch      "Thumbs.db"
001414:  C=403          9 L          24 W          213 Ch      "_src"
001631:  C=404         352 L        2504 W        53514 Ch     "access.1"
001629:  C=404         352 L        2504 W        53514 Ch     "access-log.1"
001634:  C=404         352 L        2504 W        53514 Ch     "access_log.1"
001058:  C=403          9 L          24 W          218 Ch      "typo3temp"
```

Code 200 shows a successful page. If the URL is interesting, it may be worth checking out. On the other hand, code 403 errors are almost *always* interesting because we can now tell the file exists; we just don't have access to it. As we progress in our investigation, this can give us an indication of what type of other files to look for, or it can provide us with an attack vector later on if we are able to gain access.

Based on what is discovered, we can also tailor our fuzzing strategy to go two or three levels in a particular direction.

For example, the last line of our scan detected the folder "typo3temp". Typo3 is a popular, but not as well known, CMS. Even though we do not have access to the `typo3temp` folder, we know it is there, and there is a good chance there are accessible files within that folder.

Sometimes a helpful step is to just look at the website's source code. You would be surprised at how much leftover text or comments there can be. In this case, one quick look at the code reveals that the `typo3temp` folder is still in use and should be explored further:

```
<link rel="stylesheet" type="text/css" href="https://www.biz-up.at/
typo3temp/assets/compressed/d42b6e1bdf
-bf2d65d4a223e2f396e31c35d56d6ffc.css">
<link rel="stylesheet" type="text/css" href="https://www.biz-up.at/
typo3temp/assets/compressed/flexslider
-a747cd663059c9546a6391e1d6f1a9e0.css">
```

To use Wfuzz to explore a different path within the website, we can issue the following command:

```
wfuzz -c -z file, /opt/SecLists/Discovery/Web-Content/big.txt
--hc 404,301,302 http://www.biz-up.at/typo3temp/FUZZ
```

EXPERTTIP: ALEX HEID

Even the most careful cybercriminals make mistakes. Wfuzz comes with two badass wordlists. When I run Wfuzz, I will first Wfuzz for directory names, then I will run a second fuzz for PHP, .zip, .tgz, .txt, .sql. The file 'big.txt', which comes with Wfuzz, will find most of the awesome stuff.

I regularly find files like 1.sql or Abc.sql. People making file backups rushing to make a quick filename will name the file something short and Wfuzz will just rip through and detect those. I have found source codes for entire darknet sites that way. I also look for 'c99', which is a common webshell. I find that people will use that instead of installing something like Cpanel or Plesk.

Photon

Photon is an OSINT web crawling and data extraction engine. It is highly sophisticated and a great tool to use when you are going for a wide shotgun approach. According to Photon's GitHub page, it is "an incredibly fast crawler designed for OSINT." Photon is available for download at <https://github.com/s0md3v/Photon>.

Uses for Photon include looking for creative/hard-to-find pages on a site, external links, and broken links; creating a clone of the website; and determining changes to a site over time.

There are literally hundreds of web crawlers available—many of them are as powerful as they are expensive. Photon is fantastic because it works, and it is open source.

For OSINT, Photon can easily extract different target points from a website. This can include information about staff (such as names and email addresses), contact info, related social media sites, documents, and other potential hidden company information like secret keys and bucket info.

During a typical website crawl, Photon can extract the following type of information:

- URLs (in-scope and out-of-scope)
- Parameter-based URLs
- Emails, social media accounts, buckets
- Files (pdf, doc, xls, csv, etc.)
- Secret keys (API, authentication, etc.)
- Strings matching custom regex patterns
- Subdomain and DNS related data
- Wayback/Internet archives

Photon also has built-in plugins to grab content from third-party services like Wayback Machine (Internet archive) and DNS Dumpster (for DNS-related information). There is also a `-clone` parameter that allows you to keep a local copy of the entire site.

The parameters for Photon include:

<code>-u --url</code>	root url
<code>-l --level</code>	levels to crawl
<code>-t --threads</code>	number of threads
<code>-d --delay</code>	delay between requests
<code>-c --cookie</code>	cookie
<code>-r --regex</code>	regex pattern
<code>-s --seeds</code>	additional seed urls
<code>-e --export</code>	export formatted result
<code>-o --output</code>	specify output directory
<code>-v --verbose</code>	verbose output
<code>--clone</code>	make a copy of the site
<code>--keys</code>	extract secret keys
<code>--exclude</code>	exclude urls by regex
<code>--stdout</code>	print a variable to stdout
<code>--timeout</code>	http requests timeout
<code>--ninja</code>	ninja mode
<code>--update</code>	update photon
<code>--dns</code>	enumerate subdomains & dns data
<code>--only-urls</code>	only extract urls
<code>--wayback</code>	Use URLs from archive.org as seeds
<code>--user-agent</code>	specify user-agent(s)

Crawling a Website

To perform a simple website crawl, use the `-u` switch followed by the URL:

```
python photon.py -u "http://www.vinnytroia.com"
```

Figure 7.1 shows the output of launching Photon against my personal website.



```

$ python photon.py -u http://www.vinnytroia.com
[+] URL received from robots.txt: 2
[+] URL received from sitemap.xml: 40
[+] Level 1: 42 URLs
[+] Progress: 43/43
[+] Level 2: 6 URLs
[+] Progress: 6/6
[+] Crawling 21 JavaScript files
[+] Progress: 21/21

[+] Level 3:
[+] Robots: 2
[+] Sitemap: 40
[+] Scripts: 21
[+] Javascript: 0

[+] Total requests made: 72
[+] Total time taken: 2 minutes 8 seconds
[+] Requests per second: 3
[+] Results saved in www.vinnytroia.com directory
  
```

Figure 7.1

The results saved from this command include a list of detected URLs from the target website.

At this point the `--clone` parameter will take those URLs and perform a full scrape of the site, and save every scraped page to disk:

```
python photon.py -u "http://www.vinnytroia.com" --clone
```

For more complex sites like those of big corporations, blogs, and message boards, you should modify the depth of the scan and the number of simultaneous threads. The `-t` switch allows you to specify threads, and the `-d` switch specifies the depth (meaning how many levels of nested pages will be followed):

```
python photon.py -u "http://www.site.com" -t 5 -d 5
```

Another incredibly useful feature of Photon is the ability to automatically spider DNS and related subdomains using the `--dns` switch. This will give you the ability to capture the layout of a site and all of its subdomains in one shot:

```
python photon.py -u "http://www.ethereum.org" --dns
```

Let's try it with Ethereum.org, as shown in Figure 7.2.

```

root@kali: ~# python3 photon.py -u ethereum.org -d5
[Graphical DNS Map for ethereum.org]
v1.2.1
[-] Level 1: 1 prog
[+] Program: 1/1
[-] Level 2: 1 prog
[+] Program: 1/1
[-] Crawling 1 crawler at files
[+] Progress: 1/1
[+] Level: 4
[-] Interval: 4
[-] Scripts: 1
[-] Interval: 2
[-] Crawler: 1
[+] Total requests sent: 1
[+] Total time taken: 2 minutes 2 seconds
[+] Results per second: 2
[-] Generating PNG map
[+] Results saved in ethereum.org directory

```

Figure 7.2

Photon will automatically generate a graphical DNS map of the domain using DNSDumpster.com. A typical DNS map output looks like Figure 7.3.

NOTE Photon also supports scraping and crawling archives from Wayback (the Internet archive) using the `--wayback` switch. We will talk more about Wayback and Internet archives in Chapter 8.

Intrigue.io

In Chapter 5 we discussed Intrigue.io, an automated attack surface discovery tool. Intrigue.io is an “all-in-one” type tool that does quite a bit more than just network discovery.

NOTE You can download Intrigue.io at <https://github.com/intrigueio/intrigue-core>.

For example, Intrigue’s spider module is incredibly impressive (to say the least) and requires very little effort to kick off. Using Intrigue’s GUI, let’s kick off a spider of `Ethereum.org` to see if we find anything potentially useful.

To start, select URI Spider from the task list and change the entity name to `http://www.ethereum.org` as shown in Figure 7.4. Then click Run Task to start the scan.

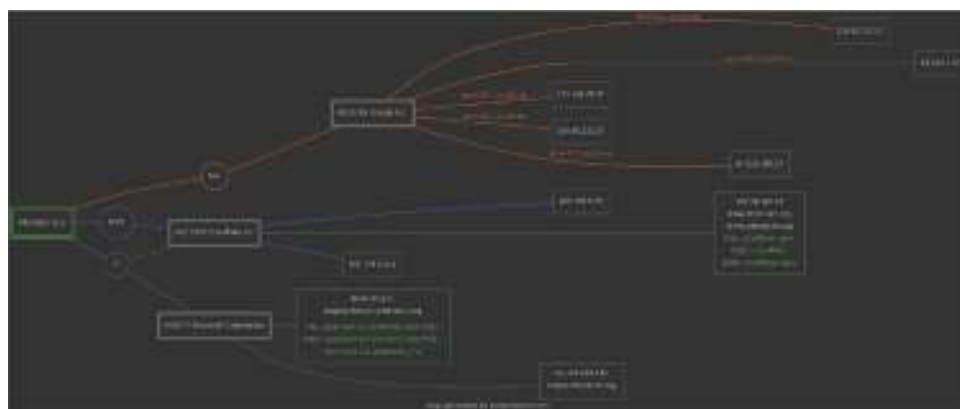


Figure 7.3

Start

Task: URI-Spider

Entity Type: Uri

Entity Name: https://www.william.org

limit: 100

max_depth: 10

spider_whitelist: (current domain)

extract_dns_records: true

extract_dns_record_pattern: (current domain)

extract_email_addresses: true

extract_phone_numbers: true

parse_file_metadata: true

extract_urls: false

user_agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_10_1)

Machine: Cmp/Root: Discovery (Active)
Machine specifies the path/process for each new entity.

Iterations: 2 iterations
Iterations specifies the count to which the machine is run.

☒ Auto Enrich

Run Task

Figure 7.4

After a short while, the Entities tab will show the results of your scan. Figure 7.5 shows a summary of the different entities initially captured.

The filter list is a very easy way to get to what you need quickly. In this case there are only 219 entities, but the ability to filter entities will become especially useful after performing large scans with hundreds of thousands of discovered entities.

To see what I mean, let's explore the EmailAddresses entities (see Figure 7.7). Clicking the link changes the table of information to display the discovered emails. In the left column we can see the emails. On the right we can see where the emails came from, which happens to be just as interesting.

email	details
["EmailAddresses:pruned@hacker.org"]	["http://hacker.org/"]
["EmailAddresses:dev@hacker.org"]	["http://hacker.org/dev/"]
["EmailAddresses:info@hacker.org"]	["http://hacker.org/info/"]
["EmailAddresses:security@hacker.org"]	["http://hacker.org/security/"]
["EmailAddresses:contact@hacker.org"]	["http://hacker.org/contact/"]
["EmailAddresses:feedback@hacker.org"]	["http://hacker.org/feedback/"]
["EmailAddresses:help@hacker.org"]	["http://hacker.org/help/"]
["EmailAddresses:support@hacker.org"]	["http://hacker.org/support/"]
["EmailAddresses:training@hacker.org"]	["http://hacker.org/training/"]
["EmailAddresses:updates@hacker.org"]	["http://hacker.org/updates/"]
["EmailAddresses:downloads@hacker.org"]	["http://hacker.org/downloads/"]
["EmailAddresses:partners@hacker.org"]	["http://hacker.org/partners/"]
["EmailAddresses:careers@hacker.org"]	["http://hacker.org/careers/"]
["EmailAddresses:privacy@hacker.org"]	["http://hacker.org/privacy/"]
["EmailAddresses:terms@hacker.org"]	["http://hacker.org/terms/"]
["EmailAddresses:about@hacker.org"]	["http://hacker.org/about/"]
["EmailAddresses:faq@hacker.org"]	["http://hacker.org/faq/"]

Figure 7.7

I consider discovered email addresses to be one of the most valuable pieces of information you can find. With corporate email addresses this may not necessarily be the case because of the sheer volume of marketing sites selling access to these emails. However, when looking for threat actors, finding a new email address can be absolute gold because it will provide you with a new pivot point on your quest to identify the actor.

Looking at the URLs, we can see download folders, subdomains, and files—in short, a slew of new information paths that can (and should) be further explored.

Summary

This chapter focused on finding hidden gems within a website by looking for hidden folders within its structure. This was accomplished by leveraging tools that try thousands of name variations in order to find working directories (i.e., bruteforcing) or that crawl a target site and gather working links based on what it finds in the site's code.

The next chapter will focus on finding hidden website data using the power of advanced search engine operators (i.e., "dorks").

Search Engine Dorks

“Dorks” are specially crafted advanced search terms that can be used on any search engine to find a wide range of publicly available information on the web. Dorks are (mostly) universal, but the examples in this book will be specifically based on Google’s search terms.

The process of dorking refers to using common error phrases that relate to a specific response code generated by a programming language. In other words, they are search queries used to find hidden (and often misconfigured) data within websites. Google Dork queries can often be used to find:

- XSS, SQLi, and other parameter-based vulnerabilities in web applications
- Confidential information from websites, such as usernames, passwords, and other forms of PII
- Online shopping info like customer data, orders, credit card numbers, and transaction numbers
- Information on printers, video cameras, and types of IOT devices

You can find an exhaustive and regularly updated list of dorks at the Exploit-DB (formerly the Google Hacking Database): <https://www.exploit-db.com/google-hacking-database>.

EXPERTTIP: ALEX HEID

Google is constantly limiting what you can search for with dork operators. You can't find as much as you used to.

Use the `site:` operator to limit Google search results to that domain name. You can do the same with Bing and DuckDuckGo. With Bing, you do this by searching an IP address—it's called the "poor man's passive DNS." It's a quick way to see everything being hosted on that IP address—domains, documents, etc.

In many instances, you can get around the Google filter performing your searches using the Google API.

Essential Search Dorks

Having even a basic understanding of search engine dorks will become a valuable tool in your day-to-day life, even outside of hacking and security investigations. This is a skill that will not only improve the quality of your search results but also help reduce your overall stress level when you can't find what you are looking for online.

This section will focus on the absolute essential search modifiers that you should learn and memorize. These are all techniques that I use on a daily basis, and you probably will, too.

The Minus Sign

I start with the minus modifier because this is probably the simplest and most important dork to learn. The minus sign (–) is what may save you from having to read through pages and pages of useless marketing and product garbage by allowing you to filter out specific terms from your search results.

This technique is amazingly useful in the right context. At the most basic level, it will help remove similar but incorrect search results that bloat queries.

Using this search modifier is as simple as it sounds: put in your search term, followed by the minus sign and whatever terms you do *not* want to appear.

Using Quotes

This search modifier is just as simple and will do the opposite of using the minus sign. Wrapping a search term in quotes will match the exact text you are looking for. For example, searching for *NSA hacker* will return results with the words NSA and Hacker, while searching for "NSA Hacker" will only return results with that exact phrase.

The site: Operator

The `site:` search operator will return search results only on the specified website. I personally find this to be one of the most useful OSINT dorks, since it allows you to focus your search on the target company.

For example, to limit your search query to a single website, you can enter the following: `site: domain.com`. When you run this query, Google will return all information related to your search specific to the “site” you are requesting.

Now that we can search for anything within a particular domain, let’s try something useful and perform a search at a specific site, like `fakebank.com`. Maybe we want a quick way to see if any Excel files (XLS) have been indexed on `fakebank.com`. We would type the following into our search engine:

```
xls site:fakebank.com
```

EXPERTTIP: ALEX HEID

To create a public attack surface, I will start with a subdomain enumeration tool like Sublist3r, which will check public databases and run wordlists against the domain to find subdomains. Then I will follow that up with Google Dorking and the `site: operator`, searching to find anything that was missed by Sublist3r.

The intitle: Operator

Using the `intitle:` operator will tell Google to show only those pages that have the specified search term in their page/HTML title. The following search term will return all sites with Pepsi in their title:

```
intitle:pepsi
```

Once you start to understand the types of titles that generic pages use, this operator becomes infinitely more useful. For example, the following search will show you pages with a publicly exposed AXIS network camera:

```
intitle:"Live View / - AXIS"
```

Another way to search for the same AXIS cameras is to combine `intitle:` operators and just look for the important keywords, like this:

```
intitle:"live view" intitle:axis
```

Figure 8.1 shows sample results of this query.

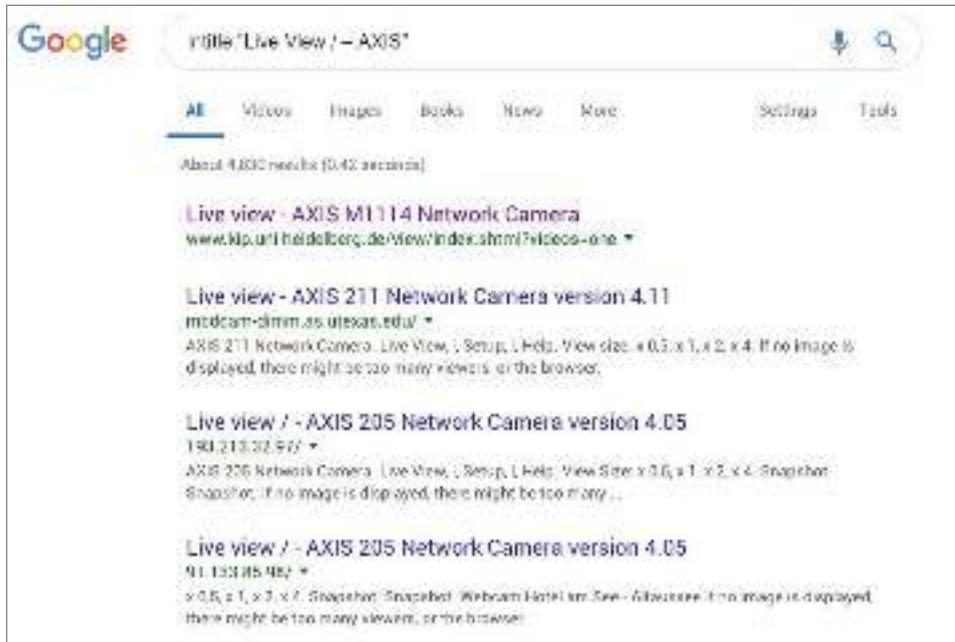


Figure 8.1

The allintitle: Operator

Rather than using the example in the previous section and having two `intitle:` operators, the `allintitle:` operator allows you to simplify your query. Google will only show you search results where all the search words are contained in the title of a page. The following search will return only results that have *all* of the words “axis,” “live,” and “view” in the title:

```
allintitle:axis live view
```

To see the differences between the results generated by the two operators, compare the results shown in Figure 8.1 with those in Figure 8.2.

The filetype: Operator

The `filetype:` operator is a good search modifier to use when looking for specific types of files that may have been forgotten, such as XLS, DOC, or CSV files. Maybe you will get lucky and find a leftover budget XLS file.

To search for a budget file on a particular site, we would use the following query:

```
Budget filetype:xls site:sitename.com
```



Figure 8.2

Looking for backup files is an incredibly important part of an investigation. People get sloppy and leave files lying around, and more often than not, such files will end in .bak:

```
filetype:bak
```

The inurl: Operator

The `inurl:` operator will give you results with your search term specifically in the URL. This does not have to be the domain name only; it can include the path or even the filename. Building on the previous example of looking for backup files, we can use this search to look for password backup files:

```
filetype:bak inurl:passwd
```

Figure 8.3 shows an example of results from this type of search.

In the mood to look for live webcams? Try this:

```
inurl:"webcam.html"
```

As shown in Figure 8.4, we can also look for Excel (XLS) files with the word “confidential” in the name:

```
filetype:xls inurl:confidential
```

Do not forget about port numbers!

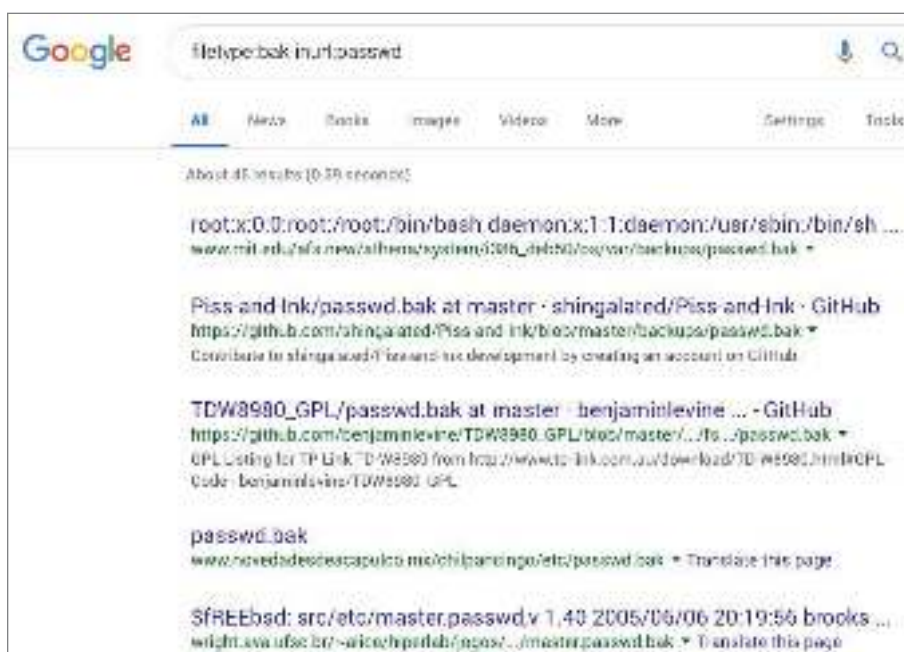


Figure 8.3

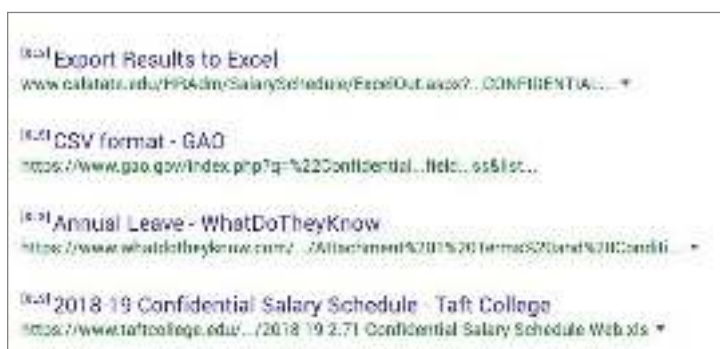


Figure 8.4

For example, if you feel like looking for active Plesk panels, you might want to search:

```
inurl:8443 plesk
```

Figure 8.5 shows example results from such a search.



Figure 8.5

The cache: Operator

The `cache:` operator is an important dork because it will let you search for pages in Google's cache. Pages and sites come and go, and sometimes it can be necessary to look for a page or file that has been removed:

```
Cache:domain.com search term
```

Searching through Google's archives and Wayback (archive.org) will be discussed in greater detail in Chapter 10.

The allinurl: Operator

Similar to `allintitle:`, the `allinurl:` operator will restrict your results to URLs that contain all of the words you specified in your search query. For example, `[allinurl: foo bar private]` will return results with "foo," "bar," and "private" in the URL. It's important to note that `allinurl:` will ignore punctuation, so `allinurl:foo/bar` will ignore the / and only return results with "foo" and "bar" in the URL.

The filename: Operator

The `filename:` extension is very similar to the `allinurl:` operator, but harder to pin down. I don't use this one as much, but this can be useful if you're looking for specific configuration backup files like `wp-config.bak`.

The intext: Operator

The `intext:` operator will search for your specific words in the text of a web page. This seems like a long shot, but don't count this operator out. This operator can be used to scan pages for *any* text you want, such as an email address, full name, PII, or even keywords from an admin login screen.

Let's use this to look for active Plesk admin panels:

```
allintext:Interface language intitle:"Plesk"
```

Figure 8.6 shows the results from this search.



Figure 8.6

The Power of the Dork

Let's take what we have learned and apply it to one of our previous sample domains: DualxCrypt.org. The majority of our network-based searches were coming up empty, but maybe we can find a nugget or two using a dorking technique.

Sometimes the most basic search can yield the biggest results. To show you what I mean, let's search for any sites that have DualxCrypt.org in the text:

```
intext:dualxcrypt.org
```

Figure 8.7 shows the results of the search.

When DualxCrypt.org was first introduced in Chapter 5, I mentioned that I learned of the domain because it was mentioned on KickAss, a Dark Web hacker forum. Even though the site clearly states it is affiliated with KickAss, the site admin (user: NSA) insisted that DualxCrypt was not constructed by him or his team.

Well, clicking the third link brings us to domainbigdata.com, and some interesting results. We can see from the results shown in Figure 8.8 that the IP is

hosted on Cypher-Net. Even if this turns out to be a dead end, this is still a pretty amazing name coincidence that cannot be ignored.

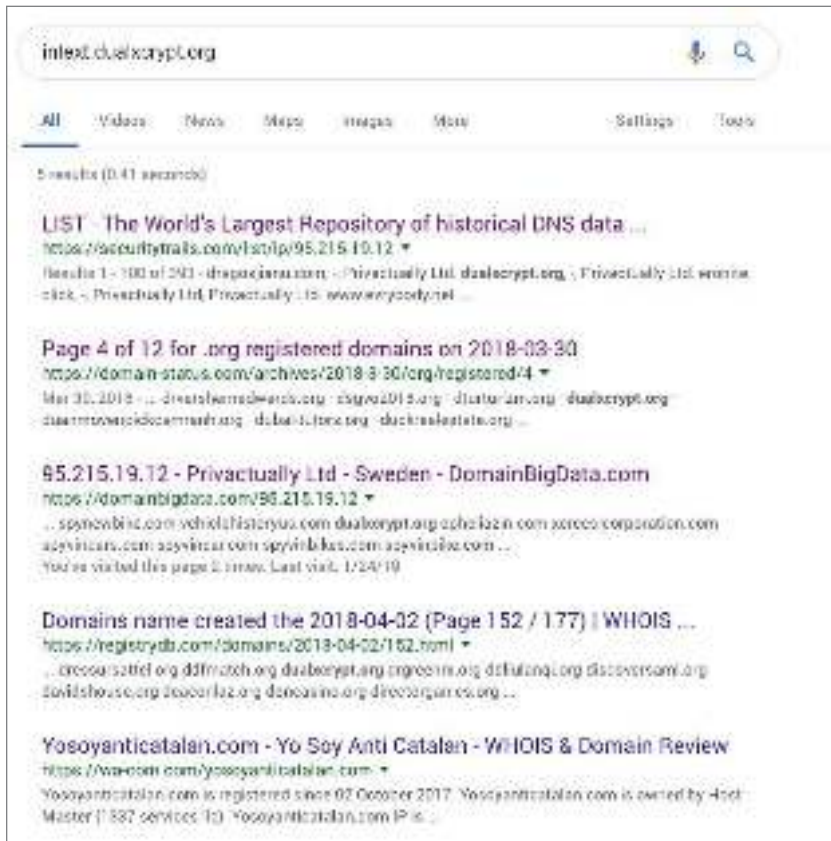


Figure 8.7

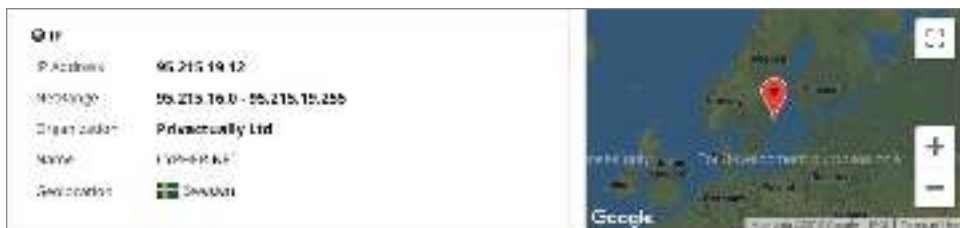


Figure 8.8

What is equally interesting is the list of other domains using the same IP address (shown in Figure 8.9).

However, in this case, they are completely unrelated. After doing some research on the IP, I discovered that it (and Cypher-Net) belong to Njalla.io, a completely anonymous domain registration company.

Njalla.io now offers anonymous web hosting as well, which explains why so many malicious sites are using that IP.

Don't Forget about Bing and Yahoo!

Google is great, but it isn't the only search engine. Oftentimes Bing will reveal search results that Google has buried. Yahoo is equally important for this reason. Each site has its own search algorithm, and if you limit your searches to one search engine you may miss important pieces of information. The dorks discussed in this chapter will work in Google, Bing, and Yahoo.

You should make it a point to run search queries in all search engines. I realize how tiring this can be, so you might want to consider automated tools to help in your quest for dork hunting across multiple search engines.

Automated Dorking Tools

Wouldn't it be great if we would take everything we have just learned about dorks and use those search queries to automate the process of vulnerability hunting or OSINT discovery across all websites?

That's exactly what Inurlbr was built to do.

Inurlbr

Inurlbr (Figure 8.10) is probably the best automated dork searching tool on the web. This tool will automate the process of searching for specific dorks across

[illegible]

Figure 8.10

multiple search engines (not just Google). The intent of this tool is to help you quickly find exploitable sites, but you can use it for other purposes as well. Inurlbr also allows you to search through TOR, Shodan Exploits, and Wikileaks. It is available for download at <https://github.com/googleinurl/SCANNER-INURLBR>.

To use Inurlbr, you provide the app with your specified dork, and it will return all the sites that it finds. This is especially useful when looking for a particular exploit, or maybe sites with a c99 shell.

Inurlbr is also an exploit finder, which this book won't discuss (and we can't publish live vulnerabilities). But some of the cool features for Inurlbr include:

- Customization of HTTP headers and user-agent strings
- Random proxy cycling
- Email and URL extraction
- Vulnerability validation
- SQLi, LFI injection exploits
- Search pages based on specific string patterns (i.e., regex)

The available switches and parameters for Inurlbr are truly exhaustive. The following code shows a few of the key ones.

This is probably only half of the actual total number of available parameters, so be sure to use the `--help` switch to view a full list:

```
-q    Choose which search engine you want through [1...24] / [e1...6]]:
      [options]:
1     - GOOGLE / (CSE) GENERIC RANDOM / API
2     - BING
3     - YAHOO BR
4     - ASK
5     - HAO123 BR
6     - GOOGLE (API)
7     - LYCOS
8     - UOL BR
9     - YAHOO US
10    - SAPO
11    - DMOZ
12    - GIGABLAST
13    - NEVER
14    - BAIDU BR
15    - YANDEX
16    - ZOO
17    - HOTBOT
18    - ZHONGSOU
19    - HKSEARCH
20    - EZILION
21    - SOGOU
22    - DUCK DUCK GO
```

```

23 - BOOROW
24 - GOOGLE(CSE) GENERIC RANDOM
-----
                SPECIAL MOTORS
-----
e1 - TOR FIND
e2 - ELEPHANT
e3 - TORSEARCH
e4 - WIKILEAKS
e5 - OTN
e6 - EXPLOITS SHODAN
-----

all - All search engines / not special motors

--proxy    Choose which proxy you want to use through the search engine:
--proxy-file Set font file to randomize your proxy to each search
            engine.

--time-proxy Set the time how often the proxy will be exchanged.
--tor-random Enables the TOR function, each usage links an unique IP.
-t          Choose the validation type:

--dork      Defines which dork the search engine will use.
--dork-file Set font file with your search dorks.
-a          Specify the string that will be used on the search script:
-m          Enable the search for emails on the urls specified.

-u          Enables the search for URL lists on the url specified.
--save-as   Save results in a certain place.
--user-agent Define the user agent used in its request against
            the target.

--regex     Using regular expression to validate his research
--replace   Replace values in the target URL.
--cms-check Enable simple check if the url / target is using CMS.
--sall      Saves all urls found by the scanner.
--ifcode    Valid results based on your return http code.
--delay     Delay between research processes.

--command-all Use this command to specify a single command to
              EVERY URL found.

```

Using Inurlbr

It should be obvious by now that Inurlbr is really meant to be a blackhat tool to scan for easily exploitable vulnerabilities. That doesn't mean it does not have legitimate and useful purposes.

TIP If you are a security entrepreneur looking to start a business or gain some security experience, you can use this tool as a foot in the door with vulnerable organizations. Once you identify the vulnerability, you can notify the organization of the bug. They might be so impressed with your work they may hire you to help secure their systems!

But be careful. You have to tell them about the vulnerability and help them close it before you ask them to take you on as a client or they might construe the gesture as a shakedown.

Inurlbr is PHP based, so running it might be a little different than the other Python scripts. Make sure you have a PHP environment set up. To start, let's search for a simple dork to search for active AXIS webcams using only Bing (results shown in Figure 8.11):

```
php inurlbr.php --dork 'inurl:view/index.shtml' -q 2 -s save.txt
```

```

[ 1 ] 7815, 188, 70, 7668, 798, 79877Cq, 7988, 180779, 48077Cq,
[ 2 ] 41, 808, C, 41, C, 42, 80, 41, 41, 80, 41, 81,
[ 3 ] 41, C 70A, C, 41, C, 85, C9, 41, 41, 47, 41, C9,
[ 4 ] 41, C -04, C, 41, C, 0081028, 41, 41774, 812049,
[ 5 ] 41, C -41, C, 41, C, 40, 7C, 41, 41, 7, 41, 7C,
[ 6 ] 41, C 781, 7C, C, 40, C8, 41, C, 41, 7, 41, C9,
[ 7 ] 7815, 188, 70, 7668, 798, 79877Cq, 7988, 180779, 48077Cq, 1812049, 181, 181, 2.1
[ 8 ]
[ 9 ]
[ 10 ]
[ 11 ]
[ 12 ]
[ 13 ]
[ 14 ]
[ 15 ]
[ 16 ]
[ 17 ]
[ 18 ]
[ 19 ]
[ 20 ]
[ 21 ]
[ 22 ]
[ 23 ]
[ 24 ]
[ 25 ]
[ 26 ]
[ 27 ]
[ 28 ]
[ 29 ]
[ 30 ]
[ 31 ]
[ 32 ]
[ 33 ]
[ 34 ]
[ 35 ]
[ 36 ]
[ 37 ]
[ 38 ]
[ 39 ]
[ 40 ]
[ 41 ]
[ 42 ]
[ 43 ]
[ 44 ]
[ 45 ]
[ 46 ]
[ 47 ]
[ 48 ]
[ 49 ]
[ 50 ]
[ 51 ]
[ 52 ]
[ 53 ]
[ 54 ]
[ 55 ]
[ 56 ]
[ 57 ]
[ 58 ]
[ 59 ]
[ 60 ]
[ 61 ]
[ 62 ]
[ 63 ]
[ 64 ]
[ 65 ]
[ 66 ]
[ 67 ]
[ 68 ]
[ 69 ]
[ 70 ]
[ 71 ]
[ 72 ]
[ 73 ]
[ 74 ]
[ 75 ]
[ 76 ]
[ 77 ]
[ 78 ]
[ 79 ]
[ 80 ]
[ 81 ]
[ 82 ]
[ 83 ]
[ 84 ]
[ 85 ]
[ 86 ]
[ 87 ]
[ 88 ]
[ 89 ]
[ 90 ]
[ 91 ]
[ 92 ]
[ 93 ]
[ 94 ]
[ 95 ]
[ 96 ]
[ 97 ]
[ 98 ]
[ 99 ]
[ 100 ]
[ 101 ]
[ 102 ]
[ 103 ]
[ 104 ]
[ 105 ]
[ 106 ]
[ 107 ]
[ 108 ]
[ 109 ]
[ 110 ]
[ 111 ]
[ 112 ]
[ 113 ]
[ 114 ]
[ 115 ]
[ 116 ]
[ 117 ]
[ 118 ]
[ 119 ]
[ 120 ]
[ 121 ]
[ 122 ]
[ 123 ]
[ 124 ]
[ 125 ]
[ 126 ]
[ 127 ]
[ 128 ]
[ 129 ]
[ 130 ]
[ 131 ]
[ 132 ]
[ 133 ]
[ 134 ]
[ 135 ]
[ 136 ]
[ 137 ]
[ 138 ]
[ 139 ]
[ 140 ]
[ 141 ]
[ 142 ]
[ 143 ]
[ 144 ]
[ 145 ]
[ 146 ]
[ 147 ]
[ 148 ]
[ 149 ]
[ 150 ]
[ 151 ]
[ 152 ]
[ 153 ]
[ 154 ]
[ 155 ]
[ 156 ]
[ 157 ]
[ 158 ]
[ 159 ]
[ 160 ]
[ 161 ]
[ 162 ]
[ 163 ]
[ 164 ]
[ 165 ]
[ 166 ]
[ 167 ]
[ 168 ]
[ 169 ]
[ 170 ]
[ 171 ]
[ 172 ]
[ 173 ]
[ 174 ]
[ 175 ]
[ 176 ]
[ 177 ]
[ 178 ]
[ 179 ]
[ 180 ]
[ 181 ]
[ 182 ]
[ 183 ]
[ 184 ]
[ 185 ]
[ 186 ]
[ 187 ]
[ 188 ]
[ 189 ]
[ 190 ]
[ 191 ]
[ 192 ]
[ 193 ]
[ 194 ]
[ 195 ]
[ 196 ]
[ 197 ]
[ 198 ]
[ 199 ]
[ 200 ]
[ 201 ]
[ 202 ]
[ 203 ]
[ 204 ]
[ 205 ]
[ 206 ]
[ 207 ]
[ 208 ]
[ 209 ]
[ 210 ]
[ 211 ]
[ 212 ]
[ 213 ]
[ 214 ]
[ 215 ]
[ 216 ]
[ 217 ]
[ 218 ]
[ 219 ]
[ 220 ]
[ 221 ]
[ 222 ]
[ 223 ]
[ 224 ]
[ 225 ]
[ 226 ]
[ 227 ]
[ 228 ]
[ 229 ]
[ 230 ]
[ 231 ]
[ 232 ]
[ 233 ]
[ 234 ]
[ 235 ]
[ 236 ]
[ 237 ]
[ 238 ]
[ 239 ]
[ 240 ]
[ 241 ]
[ 242 ]
[ 243 ]
[ 244 ]
[ 245 ]
[ 246 ]
[ 247 ]
[ 248 ]
[ 249 ]
[ 250 ]
[ 251 ]
[ 252 ]
[ 253 ]
[ 254 ]
[ 255 ]
[ 256 ]
[ 257 ]
[ 258 ]
[ 259 ]
[ 260 ]
[ 261 ]
[ 262 ]
[ 263 ]
[ 264 ]
[ 265 ]
[ 266 ]
[ 267 ]
[ 268 ]
[ 269 ]
[ 270 ]
[ 271 ]
[ 272 ]
[ 273 ]
[ 274 ]
[ 275 ]
[ 276 ]
[ 277 ]
[ 278 ]
[ 279 ]
[ 280 ]
[ 281 ]
[ 282 ]
[ 283 ]
[ 284 ]
[ 285 ]
[ 286 ]
[ 287 ]
[ 288 ]
[ 289 ]
[ 290 ]
[ 291 ]
[ 292 ]
[ 293 ]
[ 294 ]
[ 295 ]
[ 296 ]
[ 297 ]
[ 298 ]
[ 299 ]
[ 300 ]
[ 301 ]
[ 302 ]
[ 303 ]
[ 304 ]
[ 305 ]
[ 306 ]
[ 307 ]
[ 308 ]
[ 309 ]
[ 310 ]
[ 311 ]
[ 312 ]
[ 313 ]
[ 314 ]
[ 315 ]
[ 316 ]
[ 317 ]
[ 318 ]
[ 319 ]
[ 320 ]
[ 321 ]
[ 322 ]
[ 323 ]
[ 324 ]
[ 325 ]
[ 326 ]
[ 327 ]
[ 328 ]
[ 329 ]
[ 330 ]
[ 331 ]
[ 332 ]
[ 333 ]
[ 334 ]
[ 335 ]
[ 336 ]
[ 337 ]
[ 338 ]
[ 339 ]
[ 340 ]
[ 341 ]
[ 342 ]
[ 343 ]
[ 344 ]
[ 345 ]
[ 346 ]
[ 347 ]
[ 348 ]
[ 349 ]
[ 350 ]
[ 351 ]
[ 352 ]
[ 353 ]
[ 354 ]
[ 355 ]
[ 356 ]
[ 357 ]
[ 358 ]
[ 359 ]
[ 360 ]
[ 361 ]
[ 362 ]
[ 363 ]
[ 364 ]
[ 365 ]
[ 366 ]
[ 367 ]
[ 368 ]
[ 369 ]
[ 370 ]
[ 371 ]
[ 372 ]
[ 373 ]
[ 374 ]
[ 375 ]
[ 376 ]
[ 377 ]
[ 378 ]
[ 379 ]
[ 380 ]
[ 381 ]
[ 382 ]
[ 383 ]
[ 384 ]
[ 385 ]
[ 386 ]
[ 387 ]
[ 388 ]
[ 389 ]
[ 390 ]
[ 391 ]
[ 392 ]
[ 393 ]
[ 394 ]
[ 395 ]
[ 396 ]
[ 397 ]
[ 398 ]
[ 399 ]
[ 400 ]
[ 401 ]
[ 402 ]
[ 403 ]
[ 404 ]
[ 405 ]
[ 406 ]
[ 407 ]
[ 408 ]
[ 409 ]
[ 410 ]
[ 411 ]
[ 412 ]
[ 413 ]
[ 414 ]
[ 415 ]
[ 416 ]
[ 417 ]
[ 418 ]
[ 419 ]
[ 420 ]
[ 421 ]
[ 422 ]
[ 423 ]
[ 424 ]
[ 425 ]
[ 426 ]
[ 427 ]
[ 428 ]
[ 429 ]
[ 430 ]
[ 431 ]
[ 432 ]
[ 433 ]
[ 434 ]
[ 435 ]
[ 436 ]
[ 437 ]
[ 438 ]
[ 439 ]
[ 440 ]
[ 441 ]
[ 442 ]
[ 443 ]
[ 444 ]
[ 445 ]
[ 446 ]
[ 447 ]
[ 448 ]
[ 449 ]
[ 450 ]
[ 451 ]
[ 452 ]
[ 453 ]
[ 454 ]
[ 455 ]
[ 456 ]
[ 457 ]
[ 458 ]
[ 459 ]
[ 460 ]
[ 461 ]
[ 462 ]
[ 463 ]
[ 464 ]
[ 465 ]
[ 466 ]
[ 467 ]
[ 468 ]
[ 469 ]
[ 470 ]
[ 471 ]
[ 472 ]
[ 473 ]
[ 474 ]
[ 475 ]
[ 476 ]
[ 477 ]
[ 478 ]
[ 479 ]
[ 480 ]
[ 481 ]
[ 482 ]
[ 483 ]
[ 484 ]
[ 485 ]
[ 486 ]
[ 487 ]
[ 488 ]
[ 489 ]
[ 490 ]
[ 491 ]
[ 492 ]
[ 493 ]
[ 494 ]
[ 495 ]
[ 496 ]
[ 497 ]
[ 498 ]
[ 499 ]
[ 500 ]
[ 501 ]
[ 502 ]
[ 503 ]
[ 504 ]
[ 505 ]
[ 506 ]
[ 507 ]
[ 508 ]
[ 509 ]
[ 510 ]
[ 511 ]
[ 512 ]
[ 513 ]
[ 514 ]
[ 515 ]
[ 516 ]
[ 517 ]
[ 518 ]
[ 519 ]
[ 520 ]
[ 521 ]
[ 522 ]
[ 523 ]
[ 524 ]
[ 525 ]
[ 526 ]
[ 527 ]
[ 528 ]
[ 529 ]
[ 530 ]
[ 531 ]
[ 532 ]
[ 533 ]
[ 534 ]
[ 535 ]
[ 536 ]
[ 537 ]
[ 538 ]
[ 539 ]
[ 540 ]
[ 541 ]
[ 542 ]
[ 543 ]
[ 544 ]
[ 545 ]
[ 546 ]
[ 547 ]
[ 548 ]
[ 549
```

Figure 8.11

That's 164 results, and that is using only Bing! (If you want to expand your search to more than just Bing, use -q all.

The sky is really the limit here. As you dive deeper into dork searching, you will almost certainly run into search engine blocks (i.e., CAPTCHAs) to verify

that you are human. You can try to circumvent these checks by alternating your proxies.

If you are looking for dorks to search for, don't forget about the Exploit Database (<https://www.exploit-db.com>).

The `--proxy` parameter will let you specify a proxy to use, or `--proxy-file` will let you alternate proxies within a file. You can also alternate the proxies using a timed interval that you define with `--time-proxy`.

The following dork is designed to help find admin panels. It will search all search engines, use proxies within a proxy file, and alternate the proxies every second:

```
php inurlbr.php --dork 'inurl:admin intitle:login' -q all
--proxy-file proxy.txt --time-proxy 1s -s save.txt
```

The real power of Inurlbr is in its ability to test and validate LFI and SQLi vulnerabilities. That is beyond the scope of this book, but if you are interested in hunting for vulnerabilities, you should definitely give this tool a further look.

Summary

This chapter focused on using advanced search engine queries (i.e., dorks) to find information that may be otherwise hidden. This process of dorking can be used to find vulnerabilities within websites, or even to find leftover files and subdomains. This chapter covered essential dorking techniques that should be memorized and used daily.

Now that we have established a number of ways to identify subdomains and network addresses, the next chapter will focus on the cornerstone of any domain investigation: WHOIS.

The next two chapters will cover three specific topics: WHOIS, Certificate Transparency, and the Internet archive (aka the Wayback Machine). The three tools used together will be an extremely powerful investigation tool in your arsenal. This chapter focuses on the power of WHOIS data.

This chapter will provide background information on WHOIS, what it is, and why it is important, and will also look at the different services that provide WHOIS data, as well as different techniques that can be used to query and uncover historical WHOIS records.

WHOIS

WHOIS data can contain a lot of useful attribution information regarding the owner of a domain *if* you are able to access it.

The WHOIS protocol is used to query databases containing all sorts of publicly available information on Internet resources, including domain names and IP addresses. It was derived from the earlier Name/Finger Protocol, the same protocol behind the ARPANET NICNAME server, which was part of ARPANET, the precursor to the Internet.

The WHOIS protocol is used to query a wide network of WHOIS servers for any information on the domains behind the billions of websites around the world (collectively known as WHOIS data). Tons of services and tools are

available for querying WHOIS records, which can typically return information on a domain's registrant, admin, technical, and billing contacts.

Over the years, domain registration added a layer of privacy (for a fee), making it much more difficult to determine the true owner of a domain name.

In 2018, the process of attribution took an even bigger step backward with the introduction of the General Data Protection Regulation (GDPR) throughout the European Union. Soon after GDPR came out, the Internet Corporation for Assigned Names and Numbers (ICANN) voted to approve a Temporary Specification that makes most personal information related to domains unavailable to the public but allows certain parties to receive accreditation from ICANN and view a less restrictive set of WHOIS data.

NOTE This chapter will really drive home the phrase “you get what you pay for.” Open intelligence gathering would be great if all the tools and information were always completely free, but that is not always the case. Some of the most valuable and useful information you may need during your investigation will come at a premium. Perhaps you can ultimately find the information by spending hours (or days) looking through obscure sources. Either way, the price will be paid in either actual dollars or your time.

When it comes to WHOIS data, I have often found that the best data comes from sources with the largest historical archives. More often than not, newer domains will be registered using domain privacy—especially those purchased by cyber criminals/threat actors, which means a WHOIS lookup will be an immediate dead end.

However, there is a chance that an older, obscure domain name registered years ago by your target did not have the same precautions attached to it. You won't know until you go as far back as you can, and when looking for WHOIS data, that type of historical data is almost never free.

Uses for WHOIS Data

You will want to look up WHOIS data for a number of reasons. The most common reasons during an investigation are to gather new or connected information on domains involved in fraud or some other criminal activity. WHOIS can be a primary source of tracking cyber criminals, particularly if they reuse registration information for their domains.

WHOIS data can also be used to:

- Look up the registrant's contact information
- Find connected domain names, emails, and physical addresses
- Trace identities of shell corporations

- Research associations between organizations and individuals
- Identify the parties behind a domain name
- Check for the existence or availability of names (such as looking for spam addresses)

NOTE More often than not, your target will *not* be thinking five moves ahead for the days when someone will be coming after them. What I mean by that is criminals don't often start out as criminals. Their initial ventures may start out (semi) legitimate and eventually turn illegal, which means there will be more clues to find the further back you search.

It is also likely that actors will have one or two email addresses specifically designed to register domains. Once you are able to uncover those emails, you can typically find all of their domains in one shot.

Historical WHOIS

The name of the game when searching for WHOIS data on a domain is having access to the historical data. As previously mentioned, there is a good chance that the domain was not always private. There is also a small window of time during registration where the domain could have been public but then switched to private. The only way you are going to see what is really going on is with a solid historical WHOIS service.

In my humble opinion, the top two WHOIS search services are Whoisology.com and [DomainTools](http://DomainTools.com).

I have used both extensively, and for investigation purposes, I don't feel any other service comes close to offering the same level of historical data as [DomainTools](http://DomainTools.com).

However, in the interest of fairness, and because this book is supposed to focus on "open-source" intelligence tools, I will cover a few different options.

Searching for Similar Domains

Before looking up WHOIS details, we need to find target domains. There is a good chance you will already know your target's domain name, but this can be a good exercise to find new targets and expand your scope.

Namedroppers.com

[Namedroppers](http://Namedroppers.com) is a domain name search engine. It's probably one of the best free-mium tools available because the search queries act as wildcard searches.

What this means is any string you search for will appear in any part of the domain name.

For example, searching for “vinny” may show any combination of domains with the word Vinny in it:

- Vinny.com
- Vinny1.io
- MyCousinVinny.movie
- VinnyWeGetIt.org

This is significant for a few reasons. First, as I mentioned earlier, never underestimate the vanity (or lack of awareness) of a threat actor, especially in their early skid days when OPSEC was just another fancy word to learn.

Because of the site’s wide searching abilities, Namedroppers also has the added benefit of quickly finding domains related to your organization—this may include competitors or (most likely) phishing domains.

Namedroppers will show you domains that are registered and will provide you with the ability to view the WHOIS information on any of those search results.

This is ultimately a paid service, but the first 50 results are free. That may not sound like a lot, but it is enough to give you an idea of what else might be out there. If you need more than 50, for, let’s say, a customer project, the report isn’t very expensive. Let’s look at some examples.

Figure 9.1 shows the initial search windows. Searching is easy; just type your search term in the box.



Figure 9.1

Searching for USBank gives us the registered results, shown in Figure 9.2.

More than 1,000 matches is a lot. While not great for detailed investigations, this is a great tool to use if you are looking for similar domains that can potentially be used for phishing attacks.



Figure 9.2

Searching for Multiple Keywords

You can also use the search to query domains matching multiple keywords, like “bank” and “America.” When working with the search parameters, selecting Any Order will return every domain containing each keyword regardless of the order, while Exact Order will return every domain name with your search terms in the exact order in which you typed them (see Figure 9.3).

Figure 9.3

To illustrate this point, searching “America bank” with Any Order selected will return the results shown in Figure 9.4.

```

1. [WHOIS] bankofamericabankofamerica.com
2. [WHOIS] america-bank.com
3. [WHOIS] cidamerica-bank.com
4. [WHOIS] pramerica-bank.com
5. [WHOIS] chsamerica-bank.com
6. [WHOIS] chsamerica-bank.net
7. [WHOIS] america-bank.in.com
8. [WHOIS] bank-of-america-banking.com
9. [WHOIS] chsamerica-banking.com
10. [WHOIS] chsamerica-banking.net
11. [WHOIS] america-banking-online.com
12. [WHOIS] chsamerica-bankonline.com
13. [WHOIS] chsamerica-bankonline.net
14. [WHOIS] america-banks.com
15. [WHOIS] america-databank.com
16. [WHOIS] america-job-bank.com
17. [WHOIS] america-online-banking.com
18. [WHOIS] bank-of-america-online-banking.com
19. [WHOIS] bankofamerica-online-banking.biz
20. [WHOIS] bankofamerica-online-banking.com
21. [WHOIS] bankofamerica-online-banking.info
22. [WHOIS] bankofamerica-online-banking.net
23. [WHOIS] bankofamerica-online-banking.org
24. [WHOIS] bankofamerica-online-banking.us
25. [WHOIS] america-onlinebank.com
26. [WHOIS] chsamerica-onlinebank.com
27. [WHOIS] chsamerica-onlinebank.net
28. [WHOIS] america-onlinebanking.com
29. [WHOIS] bankofamerica-onlinebanking.com
30. [WHOIS] bankofamerica-onlinebanking.org

```

Figure 9.4

Now when we run the same search but using the Exact Order option, we can see a noticeable difference in the results, as shown in Figure 9.5.

In these results, we can see that the order of the search terms is preserved. Using this can make an important distinction in the quality of your results. Also, looking at the results can paint an important picture of other potential phishing sites, competitors, and sites designed to tread off your IP.


```

1. {WHOIS} 1bankofamerica.com
2. {WHOIS} 2bankofamerica.com
3. {WHOIS} 120bankofamerica.com
4. {WHOIS} aaabankofamerica.com
5. {WHOIS} abankamerica.com
6. {WHOIS} abankofamerica.com
7. {WHOIS} aboutbankofamerica.biz
8. {WHOIS} aboutbankofamerica.info
9. {WHOIS} aboutbankofamerica.us
10. {WHOIS} abuseatbankofamerica.com
11. {WHOIS} abusebankofamerica.com
12. {WHOIS} adminbankofamerica.com
13. {WHOIS} affinitybankofamerica.com
14. {WHOIS} aicportbankamerica.com
15. {WHOIS} albankofamerica.com
16. {WHOIS} alexi-bankofamerica.com
17. {WHOIS} alexis-bankofamerica.com
18. {WHOIS} allstatebankofamerica.com
19. {WHOIS} altcoinbankamerica.com
20. {WHOIS} altcoinbankofamerica.com
21. {WHOIS} alunicasabankofamerica.com
22. {WHOIS} anti-bankofamerica.biz
23. {WHOIS} anti-bankofamerica.info
24. {WHOIS} anti-bankofamerica.us
25. {WHOIS} antibankofamerica.com
26. {WHOIS} autobankamerica.com
27. {WHOIS} babankofamerica.com
28. {WHOIS} babankofamerica.com
29. {WHOIS} badbankofamerica.com
30. {WHOIS} banbankofamerica.com

```

Figure 9.5

Advanced Searches

Rather than sifting through volumes of unrelated domains, NameDroppers.com enables you to set advanced search parameters to narrow your search results, as shown in Figure 9.6.

Figure 9.6

When working with the advanced search options, selecting Starts With First Keyword will force the search results to only return domain names that start with your search term. On the flip side, selecting Ends With Last Keyword will only show domains that end with your search term (see Figure 9.7).

You can also fine-tune your results to exclude numbers, dashes, and alphacharacters, set the min and max length, and show searches based on domain extension. You get the idea.



```

1. [WHOIS] bankofamerica.biz
2. [WHOIS] bankofamerica.com
3. [WHOIS] bankofamerica.info
4. [WHOIS] bankofamerica.net
5. [WHOIS] bankofamerica.org
6. [WHOIS] bankofamerica.us
7. [WHOIS] bankofamericabankofamerica.com
8. [WHOIS] bankofamerica-aq.com
9. [WHOIS] bankofamerica-alerts.com
10. [WHOIS] bankofamerica-application.com
11. [WHOIS] bankofamerica-associate.com
12. [WHOIS] bankofamerica-bacontinuum.com
13. [WHOIS] bankofamerica-berlin.com
14. [WHOIS] bankofamerica-billing.com
15. [WHOIS] bankofamerica-bofa.com
16. [WHOIS] bankofamerica-business24-7.com
17. [WHOIS] bankofamerica-cards.com
18. [WHOIS] bankofamerica-con.com
19. [WHOIS] bankofamerica-con-activate.com
20. [WHOIS] bankofamerica-con-update-2013.com
21. [WHOIS] bankofamerica-confund.biz
22. [WHOIS] bankofamerica-confund.com
23. [WHOIS] bankofamerica-confund.info
24. [WHOIS] bankofamerica-confund.us
25. [WHOIS] bankofamerica-confund.info
26. [WHOIS] bankofamerica-confirm.com
27. [WHOIS] bankofamerica-confirm.info
28. [WHOIS] bankofamerica-confirm.net
29. [WHOIS] bankofamerica-confirm.org
30. [WHOIS] bankofamerica-continuum.com

```

Figure 9.7

Looking for Threat Actors

Now let's try a different search geared toward threat actors. A threat actor's early days are probably filled with dreams of becoming the world's greatest and most widely known hacker. A theme we will cover in future chapters is a criminal's vanity (or underlying need for affirmation).

Keeping that in mind, there is a chance that your target will have registered a domain name of their hacker alias (e.g., zerocool.com). So it's probably not a bad idea to search for your target's aliases and see if similar matching domains have already been registered.

Let's test this on our buddy Cyper, as shown in Figure 9.8.

There are a lot of potential domains worth exploring. Clicking the WHOIS button next to each domain will take you to GoDaddy's site to perform a WHOIS lookup.

I don't personally find this to be useful, but I also don't want to discount it as a first search option. Depending on your goal this might be enough. However, more often than not domains will have privacy enabled and you will see something like this:

```

Admin Name: Registration Private
Admin Organization: Domains By Proxy, LLC
Admin Street: DomainsByProxy.com

```

```

Matched 717 domain names out of 170,709,443 active records!
1. [WHOIS] cyper.com
2. [WHOIS] cyper.net
3. [WHOIS] cyper.org
4. [WHOIS] cyper.us
5. [WHOIS] blockcyper.com
6. [WHOIS] hardcorecyper.com
7. [WHOIS] justyhardcorecyper.com
8. [WHOIS] justylatincyper.com
9. [WHOIS] persianascyper.com
10. [WHOIS] scyper.com
11. [WHOIS] shascyper.com
12. [WHOIS] threadcyper.com
13. [WHOIS] ventascyper.com
14. [WHOIS] cyper-demo.com
15. [WHOIS] cyper-digital.com
16. [WHOIS] cyper-digital.info
17. [WHOIS] cyper-digital.net
18. [WHOIS] cyper-german.com
19. [WHOIS] cyper-hip.net
20. [WHOIS] cyper-hc.com
21. [WHOIS] cyper-hub.net
22. [WHOIS] cyper-meal.com
23. [WHOIS] cyper-monday.com
24. [WHOIS] cyper-sa.com
25. [WHOIS] cyper-space.com
26. [WHOIS] cyper-stuhl.com
27. [WHOIS] cyper-xp.com
28. [WHOIS] cyper18.net
29. [WHOIS] cyperx.com
30. [WHOIS] xucypera.com

```

Figure 9.8

This is the type of response you will likely see with standard WHOIS searches. All of my personal domains have private registration, so I would be surprised to find a domain belonging to a threat actor that does not have privacy enabled.

Whoisology

Whoisology (www.whoisology.com) is a domain ownership archive focused on cybercrime investigations, corporate intelligence, and legal research. In short, Whoisology is a service designed to provide you with detailed information regarding a domain's registration history.

Figure 9.9 shows the initial Whoisology search page.

Let's start our research by investigating the owner of *TheDarkOverlord.com*. The likelihood of this matching to anything useful is slim to none, but we at least have to try. Figure 9.10 shows the results of searching Whoisology for *TheDarkOverlord.com*.

Current WHOIS data shows us private registration and (unfortunately) nothing useful. That is the beauty of sites that provide *historical* WHOIS information. You can go back several years to try to find attribution to the domain owner. Figure 9.11 shows the historical options available for *TheDarkOverlord.com*.

Figure 9.9

Admin Contact		Other Details	
The Admin Contact is the person or organization who controls the domain.		These are technical details & related, connected to the domain.	
Name	Admin Email: Select Contact Domain Holder Link: https://www.godaddy.com/whoisresults.aspx?domain=FREDARKOVERLORD.COM [1] Changes: +0 ccTLD: 0	Registrar Name	GoDaddy.com, LLC [53,874,567] Changes: +5,817,244 ccTLD: 1,389,542
Org		Created Date	2014-01-11 [1,107,011] Changes: +836 ccTLD: 2,047
Email		Whois Servers	whois.godaddy.com [55,166,630] Changes: +4,328,552 ccTLD: 1,329,010
Screen 1		Updated Date	2018-08-13 [1,191] Changes: +1,221 ccTLD: 567
Screen 2		Expires Date	2019-03-11 [371,511] Changes: +23,014 ccTLD: 42,044
City		Name Servers	NS1.KNOXWHOST.COM [36] Changes: +1 ccTLD: 1 NS2.KNOXWHOST.COM [38] Changes: +2 ccTLD: 1
Region		Archive Date	2018-10-30
Zip / Post			
Country	UNITED STATES [66,457,810] Changes: +17,487,545 ccTLD: 945,140		

Figure 9.10

Starting with the results from 2016, we immediately find a domain owner (Figure 9.12).

NOTE Important note: I do not believe www.darkoverlord.com has anything to do with the actual hacking group. I was just using that as an example.

In order to illustrate the differences between services, let's perform one additional WHOIS lookup on fellow security researcher whitepacket.com. Figure 9.13 shows the initial results.

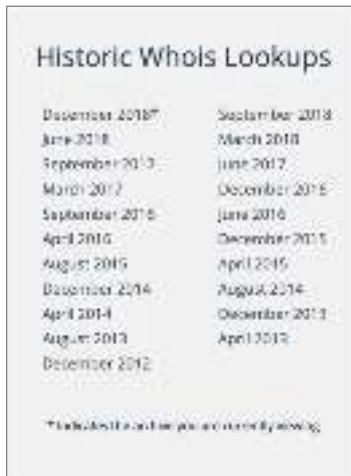


Figure 9.11



Figure 9.12

The initial results are protected and there are no connected domains. This is to be expected for an initial WHOIS lookup since we have not gone into any

of the historical data. With Whoisology, we can use the side Historic Whois Lookups menu (Figure 9.14) to go back further in time.

Registrant Contact		Connected Domains	
The Registrant Contact is the person or organization who legally owns the domain.		No data available for comparison.	
Name	Whois Agent (1,358,172) Changes: +16,481 -01/01/2014		
Org.	Domain Protection Services, Inc. (811,040) Changes: +210,001 -07/01/2013		
Email			
Street	PO Box 1769 (615,113) Changes: +4,998 -01/01/2014		
Street 2			
City	Denver (514,521) Changes: +815,790 -07/01/2013		
Region	CO (5,670,454) Changes: +97,029 -01/01/2014		
Zip / Post	80201 (615,002) Changes: +74,903 -07/01/2013		
Country	UNITED STATES (79,652,002) Changes: +12,618,758 -07/01/2013		
Phone	1 720809072 (615,032) Changes: +19,808 -01/01/2014		
Fax	1 720809072 (615,032) Changes: +75,300 -07/01/2013		

Figure 9.13

Historic Whois Lookups	
December 2018*	September 2018
June 2018	March 2018
December 2017	September 2017
June 2017	March 2017
December 2016	September 2016
June 2016	April 2016
December 2015	August 2015
April 2015	December 2014

* Follows the month you currently viewing

Figure 9.14

Looking at the historical table in Figure 9.14, we can see WHOIS records going as far back as December 2012. We will dig more into the historical values later.

For now, the important thing to note is that we have many options for historical attribution.

Advanced Domain Searching

Whoisology also has a really nice advanced search feature where you can search for strings in any part of the domain (similar to Namedroppers.com).

As they put it, you can use their advanced search to find “digital needles in over 100,000,000 haystacks” by using “multiple filters to sift through an impossible amount of data.” In other words, it’s really useful.

Running a quick advanced search for “cyper” gives us a slew of results that we can potentially investigate (Figure 9.15).

NOTE Unfortunately, none of these domains turned out to be useful. I am using Cyper as our threat actor search parameter for consistency, but also to reinforce the fact that not every search or tool will yield useful results. How realistic would it be if every example I use turned out to have a goldmine of useful information? It is never that simple . . . but in this example, we can still see the domain results and how they could *potentially* be useful.

Archive: December 2018 Must Contain - Anywhere - cyper		
polityperformance.com	cypermesatop	stacyperaltaz-box.com
polityperscope.com	manycyperformance.com	macypertf.com
lucypereddbv.com	belcyperseals.com	slateagencypermtting.com
lucyperes.com	cypermiltown.com	privacyperiod.com
residenypermits.com	cyperspace.com	lucyperry.com
residenypersonalstatements.com	cyperfastgheter.com	danypensins.com
scyper.com	cyperformingarts.com	webagencyperugis.com
cypercom.work	cyperel.com	cyperlaw.com
cypermen	askcyper.com	cyperis.com
agencyperiscope.com	minimacyper.com	legacyperpersonal.com
belcyperfirmagnt.com	personalityperformantall.com	legacyperformancescoaching.com
cyperol.org	plumacyper.com	soundswamkondcyper.com
cyper.org	stacyperaffirms.com	cryptocurancyperformance.com
cyperusmedia.org	stacyperny.com	macrosolycyperspectives.com
digitalagencyperth.org	stacyperman.com	julicycyperfume.com

Figure 9.15

Worth the Money? Absolutely

Whoisology is an excellent service that provides an affordable option for accessing a significant amount of historical domain history. Whoisology also offers rich API connections to many of the tools previously discussed, such as Intrigue.io, SpiderFoot, and more. If you are looking for an affordable option (or do not want to pay a monthly fee), or something that easily plugs into your existing toolsets for the fastest possible results, this may be your way to go. Whoisology works on credits—it is \$5 per domain report—so you can buy only what you need without the monthly overhead. This is very useful when working with tools like Intrigue—you can just use/buy credits as you need them.

DomainTools

DomainTools (DT) can be your one-stop shop for everything domain-related—if you are willing to pay for it. A great tool can very easily become your “third arm,” and for me, this is one of those tools. If I were to prioritize the importance and necessity of tools discussed in this book, DomainTools would absolutely be in my top three.

Domain Search

DomainTools offers its own version of the broad keyword-based searches discussed earlier in this chapter. In addition, DT will tell you whether the domain is registered. Figure 9.16 shows the initial DT domain search screen.

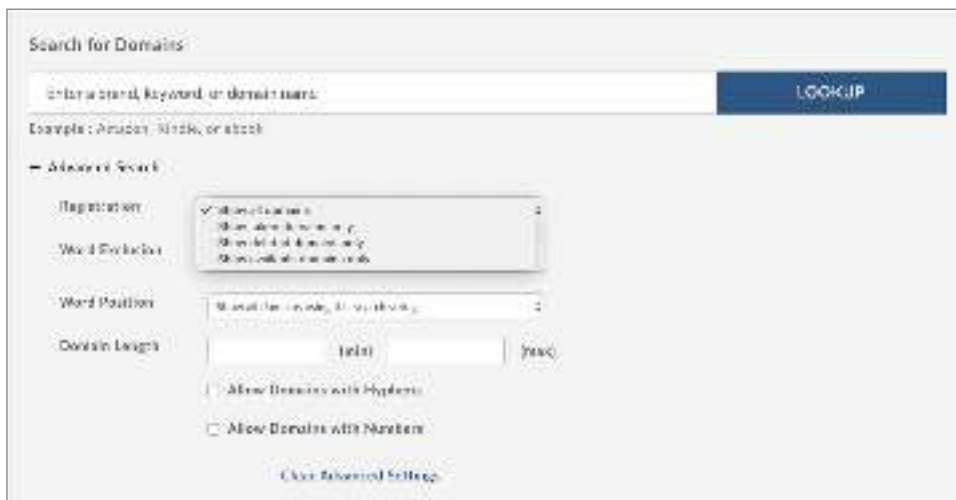
The screenshot shows the 'Search for Domains' interface. At the top, there is a text input field labeled 'Enter a word, keyword, or domain name:' and a blue 'LOOKUP' button. Below the input field, an example 'Example: Amazon, Kindle, or ebook' is provided. A section titled 'Advanced Search' is expanded, showing several filters: 'Registration' with a dropdown menu (currently showing 'All domains'), 'Word Exclusion' with a text input field, 'Word Position' with a dropdown menu (currently showing 'Word in the middle'), and 'Domain Length' with two input fields for 'min' and 'max'. There are also two checkboxes: 'Allow Domains with Hyphens' (checked) and 'Allow Domains with Numbers' (unchecked). At the bottom of the search section, there is a link that says 'Click Advanced Settings'.

Figure 9.16

Bulk WHOIS

Bulk Parsed WHOIS accepts a list of domains via the text input window and automatically generates and downloads a CSV file containing parsed WHOIS records for those domains. You can enter up to 2,000 domains at a time, and you can run successive queries up to your allotted monthly query limit.

This is a useful feature if you need to get the results of multiple domains at once, and in a nice, easy-to-read CSV format.

Reverse IP Lookup

Having a reverse IP lookup tool here comes in handy. It saves you the time of having to go look someplace else, and DT's historical records can give you results that you may not find with another tool. As a quick example, let's look up 192.241.180.214, which is the IP address of my site. (See Figure 9.17)

Figure 9.17

We can see from the results shown in Figure 9.18 that a handful of domains are active on that IP address. This can often be a huge find, especially when looking into threat actors. There is a good chance an actor will have multiple sites hosted on the same server/IP address.

Reverse IP Lookup Results — 6 domains hosted on IP address 192.241.180.214

[Download 6 results as .CSV](#)

Domain	View Whois Record	Screenshots
1. survive.com	☐	
2. survive.net	☐	
3. survive recordings.com	☐	
4. tighillon.net	☐	
5. nightlionsecurity.com	☐	
6. winmywhois.com	☐	

Figure 9.18

From here, you can click View Whois Record to see the WHOIS record for that account.

WHOIS Records on Steroids

To illustrate the power of DomainTools, let's search for our fellow security researcher WhitePacket. Figure 9.19 shows the initial result for a WHOIS lookup on WhitePacket.com.

Whois Record for WhitePacket.com	
— Domain Profile	
Proximity Score	33
Email	abuse@name.com is associated with ~1,913,560 domains
Registrar	Name.com, Inc. IANA ID: 625 URL: http://www.name.com Whois Server: whois.name.com abuse@name.com Tel: 1/2021101849
Registrar Status	Client Transfer Prohibited
Dates	1204 days old Created on 2015-10-18 Expires on 2019-10-18 Updated on 2018-10-18
Name Servers	RAFF.NS.CLOUDFLARE.COM (has 11,031,416 domains) ROCKY.NS.CLOUDFLARE.COM (has 11,031,416 domains)
IP Address	104.24.118.111 - 517 other sites hosted on this server 104.24.119.111 - 488 other sites hosted on this server
IP Location	🇺🇸 - California - San Francisco - Cloudflare Inc.
ASN	🇺🇸 AS13335 CLOUDFLARENET - Cloudflare, Inc., US (registered Jul 14, 2010)
Domain Status	Registered And Active Website
Whois History	57 records have been archived since 2006-01-31
IP History	24 changes on 24 unique IP addresses over 15 years
Registrar History	4 registrars with 2 drops
Hosting History	14 changes on 10 unique name servers over 16 years
Whois Server	whois.name.com
— Website	
Website Title	🌐 WhitePacket Home
Server Type	cloudflare
Response Code	200
SEO Score	95%
Tags	4/5 (Unique: 25/, Linked: 23)
Images	4 (Alt tags missing: 4)
Links	11 (Internal: 4, Outbound: 7)

Figure 9.19

A few things are worth mentioning here:

1. We can immediately see that the name servers are with Cloudflare. If you are trying to trace a website to its actual server, this can be a giant dead end as people (and cyber criminals) will often mask their website behind Cloudflare's DNS. We will look at ways to get around this in the next chapter (Certificate Transparency).
2. DomainTools has 67 changes of WHOIS history going back to 2006, whereas Whoisology only went back as far as 2012.
3. Over the past 15 years, there have been 34 unique IPs tied to this domain. As our investigation continues, finding the hosting provider or server attached to the domain may be important if we are trying to track the actual location of the IP.

The second column of information displayed on the DomainTools page provides a gateway to expanding on these searches (Figure 9.20).



Figure 9.20

Right away we can see that DT provides a screenshot history. This will be relevant as we track the different owners through the WHOIS history. Different owners will have different websites, and different websites will offer different clues.

This is a huge time saver so we don't have to go digging around Wayback/Archive.org (which we will discuss in the next chapter).

WHOIS History

Looking at the WHOIS history for `WhitePacket.com`, we can see (by the little eye mark shown in Figure 9.21) that the registration information became private on October 18, 2015. This is where the historical data really becomes a pivotal factor in a potential investigation. Having access to a source of data that can show us who the registration belonged to *before* the registration became private can be a monumental discovery. The next steps would be to go as far back as possible to try to determine an owner. If that doesn't work, we can use other tactics to identify a site owner.

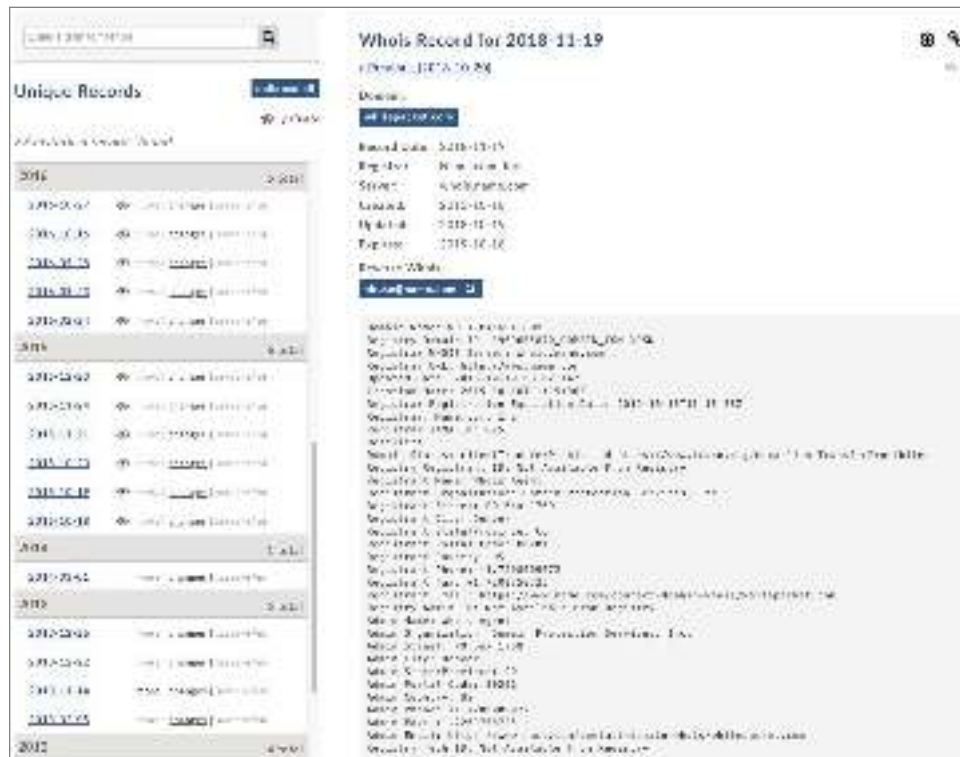


Figure 9.21

The Power of Screenshots

As a hypothetical, let's say we were only interested in the domain activities of WhitePacket.com around the year 2005. In this particular case, we *can* see the registration data from that year, but let's *pretend* that this is not the case and the registration is private. We would be up Schitt's Creek, right?

This is why it pays to look at a website's historical data. You never know what clues you will find written in the archives. In the case of DomainTools, it provides site screenshots for you, so you don't have to do any additional digging.

They say a picture is worth a thousand words. In this case, it's worth at least two: the first and last name of the website's owner in 2005, written right on the page (Figure 9.22).

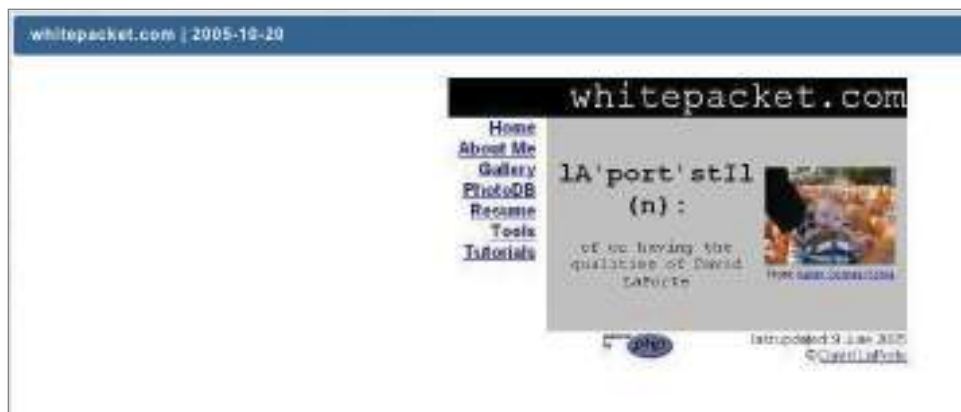


Figure 9.22

This is a perfect example of why the Wayback Machine (Archive.org) is a critical tool for an investigation (which we will cover more in-depth in the next chapter). For now, the DomainTools' screenshot feature has given us the owner of WhitePacket.com circa June 9, 2005.

Digging into WHOIS History

The screenshot example was extremely lucky. Chances are you won't find an answer so obvious with the owner's name literally written on the website. However, there is a lot of very useful information to be gained from looking back into a domain's historical WHOIS data. Let's continue on exploring WhitePacket.com to understand why.

Looking for Changes in Ownership

In the previous example, we saw that the majority of the WHOIS records are blocked with private registration (Figure 9.23).

The screenshot displays a WHOIS record for the domain **www.mad4u.com**. The interface is divided into two main sections: a left sidebar for navigation and a main content area for the record details.

Left Sidebar:

- Unique Records:** A list of records grouped by year. The years shown are 2015 (3 total), 2014 (3 total), and 2013 (3 total). Each year has a link to view more records.

Main Content Area:

- Whois Record for 2015-11-19:** The title of the selected record.
- Previous:** A link to view the previous record.
- Domain:** **www.mad4u.com**
- Registrar:** **GoDaddy.com, Inc.**
- Creation Date:** **2013-11-19**
- Expiration Date:** **2015-11-19**
- Updated Date:** **2015-11-19**
- Domain Name:** **www.mad4u.com**
- Registrar:** **GoDaddy.com, Inc.**
- Creation Date:** **2013-11-19**
- Expiration Date:** **2015-11-19**
- Updated Date:** **2015-11-19**
- Domain Name:** **www.mad4u.com**
- Registrar:** **GoDaddy.com, Inc.**
- Creation Date:** **2013-11-19**
- Expiration Date:** **2015-11-19**
- Updated Date:** **2015-11-19**

Figure 9.23

Looking through the historical list, we can see that the last nonprivate registration data for the domain was available on March 1, 2014 (Figure 9.24).

The screenshot displays a historical list of WHOIS records for the domain **www.mad4u.com**. The interface is divided into two main sections: a left sidebar for navigation and a main content area for the record details.

Left Sidebar:

- Unique Records:** A list of records grouped by year. The years shown are 2015 (3 total), 2014 (3 total), and 2013 (3 total). Each year has a link to view more records.

Main Content Area:

- Whois Record for 2015-11-19:** The title of the selected record.
- Previous:** A link to view the previous record.
- Domain:** **www.mad4u.com**
- Registrar:** **GoDaddy.com, Inc.**
- Creation Date:** **2013-11-19**
- Expiration Date:** **2015-11-19**
- Updated Date:** **2015-11-19**
- Domain Name:** **www.mad4u.com**
- Registrar:** **GoDaddy.com, Inc.**
- Creation Date:** **2013-11-19**
- Expiration Date:** **2015-11-19**
- Updated Date:** **2015-11-19**

Figure 9.24

NOTE Just a friendly reminder that all names, addresses, phone numbers, and emails will be (mostly) changed in the examples. If you are trying this at home, your results may be different.

We can see there is a huge gap of time (over a year) between when the domain registration was open on March 1, 2014, and when it became private on October 18, 2015. This could potentially indicate a change in ownership, possibly through a lapse in ownership (i.e., the domain expired). This is something we can (and should) confirm.

After we click the 2014 domain result, Figure 9.25 shows that the domain was owned by Johnny Framperson of 178 Westbury Court, London, England.

```
Domain Name: WHITEPACKET.COM
Registry Domain ID:
Registrar WHOIS Server: whois.freeparking.co.uk
Registrar URL:
Updated Date: 05-Feb-2014
Creation Date: 25-Dec-2013
Registrar Registration Expiration Date: 25-Dec-2015
Registrar: Freeparking Domain Registrars Ltd
Registrar T&A ID: 000
Registrar Abuse Contact Email:
Registrar Abuse Contact Phone:
Domain Status: REDEMPTIONPERIOD
Registry Registrant ID: 07 20004244
Registrant Name: Domain Contact 14289461
Registrant Organization: Johnny Framperson
Registrant Street: 178 Westbury Court
Registrant City: London
Registrant State/Province: England
Registrant Postal Code: E11C 6RL
Registrant Country: GB
Registrant Phone: +44 2056769121
Registrant Phone Ext:
Registrant Fax:
Registrant Fax Ext:
Registrant Email: johnnyboy@fakemail.com
Registry Admin ID: 01_20004244
```

Figure 9.25

Looking at the Domain Status field, we can see that the domain was in “REDEMPTIONPERIOD,” indicating that Johnny let the domain expire around this time. This means the current owner of WhitePacket.com is probably not the same person who owned it in 2014.

While it is technically possible that the 2014 owner could have let the domain expire and then purchased it back up in 2015 with private registration status, that scenario seems unlikely.

However, this is something to keep in the back of your mind. Many reasons exist for why someone would let a domain expire, then buy it back up again using private registration. For example, maybe this threat actor is so smart that he knew he wanted to use the domain for malicious purposes and anticipated this type of future investigation. Again, highly unlikely, but you never know.

Reverse WHOIS

We now have several new pieces of information on the possible owner of WhitePacket.com dating back to 2014. This is where the Reverse WHOIS search can come into play. The Reverse WHOIS search engine lets us look up domain ownership (in reverse) by searching for a person's name, company phone number, physical address, or email address. The results of your search will be a list of domain names that have your search term somewhere in their WHOIS records. Figure 9.26 shows the DomainTools reverse WHOIS search box.

Figure 9.26

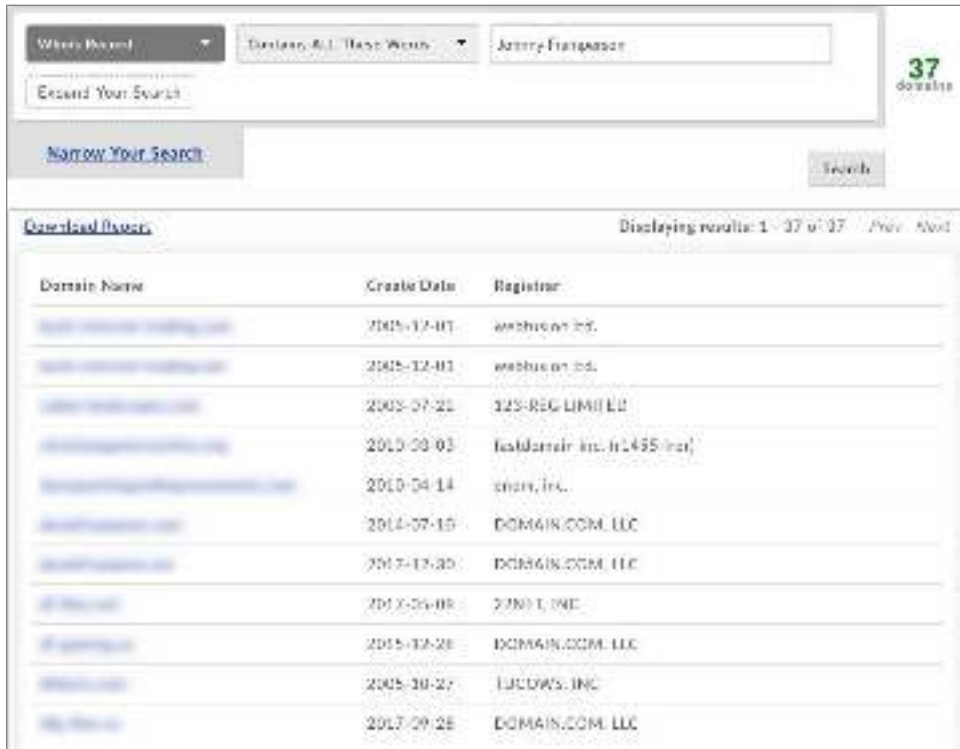
Let's see if we can find any additional domains owned by our new lead, Johnny Framperson. Figure 9.27 shows the results of a search for a WHOIS record containing "all of the words" Johnny and Framperson.

Domain Name	Create Date	Registrar
domaintools.com	2017-12-18	DOMAIN.COM, LLC
framperson.com	2018-12-18	DOMAIN.COM, LLC
johnnyframperson.com	2017-09-18	DOMAIN.COM, LLC

Figure 9.27

The results came back with three active domains (that we can't show you, sorry!). Not a bad find! We now have three new and *active* domains also owned by Mr. Framperson. The default results will only show us domains that are active *and* owned by the person in our search term.

That's great, but did you notice the link on the right column of Figure 9.27 that says "Add history and get:"? This will add any domains previously owned by him as well! Figure 9.28 shows the new results when we click that link.



Domain Name	Create Date	Registrar
johnnyframperson.com	2015-12-01	webhosting.com
johnnyframperson.net	2015-12-01	webhosting.com
johnnyframperson.org	2015-07-21	123-REG LIMITED
johnnyframperson.info	2015-09-03	hostdomain inc. h2495 (nat)
johnnyframperson.biz	2015-04-14	name, inc.
johnnyframperson.us	2014-07-15	DOMAIN.COM, LLC
johnnyframperson.co	2017-12-25	DOMAIN.COM, LLC
johnnyframperson.me	2017-06-08	27011, INC.
johnnyframperson.io	2015-12-28	DOMAIN.COM, LLC
johnnyframperson.tv	2015-10-27	1111111, INC.
johnnyframperson.mobi	2017-09-28	DOMAIN.COM, LLC

Figure 9.28

Boom! We now have 37 domains previously owned by this person. That's a lot. I hope you are keeping good notes and a good tracking system because this will mean a lot of additional tangent investigations to find more clues! We will investigate a few of these domains further in the next chapter.

Cross-Checking All Information

I can't tell you how many times I have forgotten to do this. This takes discipline and a tremendous amount of organization. If you are investigating a threat actor, getting into this mode of documenting and storing all clues in an organized system will quickly pay off.

Remember, most (if not all) threat actors have multiple aliases and personas. As you start to get into the weeds with researching a particular alias, you will most likely forget clues you have found elsewhere. A good system of documentation will solve this dilemma. Let me show you what I mean.

When we searched Johnny Framperson's name, we found 37 other domains that he either currently owns or owned at one point. Threat actors will rarely use their own name, but there is a good chance that they will reuse other pieces of information like an address, phone number, or email address. To illustrate

my point, Figure 9.29 shows the results of a reverse WHOIS search for Johnny Framperson's email address, johnnyboy@iMadeThisUp.com.



The screenshot shows a web interface for a reverse WHOIS search. At the top, it says "Download Report" on the left and "Displaying results: 1 - 2 of 2" on the right. Below this is a table with three columns: "Domain Name", "Create Date", and "Registrar". The first row shows a domain (partially obscured by a blue highlight), a creation date of "2014-07-10", and a registrar of "DOMAIN.COM, LLC". The second row shows "whitepacket.com", a creation date of "2015-10-18", and a registrar of "NAME.COM, INC.". At the bottom of the table, there is a link that says "Download Report" and another "Displaying results: 1 - 2 of 2" on the right.

Domain Name	Create Date	Registrar
[Redacted]	2014-07-10	DOMAIN.COM, LLC
whitepacket.com	2015-10-18	NAME.COM, INC.

Figure 9.29

Because our search results dropped from 37 to 2, we can conclude that our target uses different email addresses for domain registration. As part of a normal investigation, we would need to look through each of the other domains to find which email address(es) he uses. Once we have those, we document them, and perform more searches to find additional domains that can potentially be registered under those emails that we may have missed. As you can tell, the entire process can quickly turn into one giant rabbit hole.

NOTE This is when using a mind mapping tool will really come in handy. If you don't have a method for notating and expanding each of these different trails, I highly suggest looking into a mind mapping tool. Plenty are available. Mindnode is a great tool for Mac users. If you prefer an online app, Coggle.it or Lucidchart are both good. You have so many to choose from. Just pick something, because writing everything down in OneNote probably isn't going to cut it.

I can guarantee you will end up losing track of information or overlooking important clues—because I have. Many times. Save yourself the pain of having to go back and re-research data points and get into the habit of properly documenting things from the start. And take screenshots of *everything* (another lesson I had to learn the hard way).

Getting back to our search, let's go back and run a reverse WHOIS search on 178 Westbury Court, which is the mailing address that was listed in the registration details for WhitePacket.com. Figure 9.30 shows the results of the reverse address search.

And look at that! We have just discovered an additional domain name that did not come up in any of our previous searches. See how this works? It is a lot of cross-checking facts and data points, but it pays off!

More importantly, we can see the obvious power of being able to search historical WHOIS archives. We will dig into this topic further in the next chapter.



The screenshot shows a web interface for a Whois report. At the top, there is a link 'Whois Report' and a status 'Displaying results: 1 - 3 of 3'. Below this is a table with three columns: 'Domain Name', 'Create Date', and 'Registrar'. The table contains three rows of data. At the bottom, there is another 'Whois Report' link and a status 'Displaying results: 1 - 3 of 3'.

Domain Name	Create Date	Registrar
domain.com	2014-07-10	DOMAIN.COM, LLC
example.com	2014-08-15	SYNADOM LLC/DOMAIN, LLC
website.com	2015-10-18	NAME.COM, INC

Figure 9.30

Summary

This chapter covered the basics of WHOIS searching, including several free and premium services that can be used to identify domain ownership. Not all services are created equal, and this is one area where you will most likely end up having to spend money to find the best results. Unfortunately, there are no great tools or solutions for querying free historical WHOIS data, and the further back you want to search, the more a service will cost you.

This chapter also covered using these tools to perform reverse WHOIS queries that can enable us to find additional clues of domain ownership, which may eventually lead you to full owner attribution. Whoisology.com is a great, low-cost historical domain search tool, charging only \$5 per report. On the other hand, DomainTools offers a more premium service with considerably better (and more complete) historical records, at a much higher premium price.

In the next chapter, we will cover using certificate transparency logs to identify domain owners, using Internet archives such as the Wayback Machine.

Certificate Transparency and Internet Archives

In this chapter, we are going to focus on two main topics: Certificate Transparency and Internet archives (e.g., Wayback Machine and search engine caches). We will look at how they can all be used to further an investigation of a site's owner through the use of WHOIS data, or by being able to look at and reference older copies of a site.

Certificate Transparency

Certificate Transparency (CT) is a Google-led, open-source effort to provide auditing and monitoring of TLS/SSL server certificates issued by Certificate Authorities (CAs). The general idea behind CT is to mitigate the threat of malicious hackers impersonating a website (i.e., man-in-the-middle [MITM] or phishing attacks) due to use of stolen or forged SSL certificates.

Prior to Certificate Transparency, whenever a user visited a website using a fake SSL certificate, the site appeared “normal” to the web browser, which had no way to determine whether the SSL certificate was valid (or generated by the correct party).

In one case, a prominent Dutch Certificate Authority (DigiNotar) was compromised, allowing hackers to use their systems to generate fake SSL certificates. Those certificates were used to impersonate a number of sites including Gmail

and Facebook. Because the phishing sites appeared legitimate, the MITM attacks were successful, allowing the hackers to spy on users and steal their information.

NOTE A man-in-the-middle attack occurs when a hacker is able to intercept traffic between the victim and their destination website. One example would be if someone set up a fake WiFi hotspot at your local coffee shop. People connect to the “free WiFi” access point (sometimes automatically), not realizing it belongs to the person in the back of the shop. Now when the victim communicates with a website, all of the traffic is passing through the hacker’s WiFi access point. In theory, the hacker can read all of the traffic, which means they can steal credit card information, personal details, passwords, etc.

In these scenarios, the web browsers see nothing wrong with the certificate because the CA appears to be in good standing. This gives the users the impression they are visiting an authentic website, when they are actually being phished.

To mitigate this threat, Google established the CT project in 2015. Since then, all four major browsers—Chrome, Firefox, Opera, and Safari—require websites to have certificates with signed certificate timestamps (SCTs), which proves that the certificate has been submitted to a log. If the website you’re trying to visit does not have an SCT, the browsers will warn you that it might be potentially insecure.

What Does Any of This Have to Do with Digital Investigations?

Since the CT project makes certificates publicly viewable, anyone interested in knowing about a specific domain can just open it up in a browser and view the certificate information. Because this publicly available information includes information about the owner (typically either a person or an organization), anyone who accesses the certificate can not only verify the certificate’s owner, but *can also view other sites using the same certificate*.

When you consider this point from the perspective of wildcard certificates (which let you certify any subdomain using a single certificate) or sites that share certificates across multiple websites, this can be a very significant way to research the existence of other sites.

Many of the tools discussed in previous chapters have built-in integrations to use Certificate Transparency to look for subdomains and related domains. You can use those if you want. The rest of this chapter will show you how to use tools specifically designed to search through CT logs. Let’s dig in.

Scouting with CTFR

CTFR is an open-source, command-line tool that allows identification of the subdomains under a particular domain by accessing Certificate Transparency logs.

You can download CTFR at <https://github.com/UnaPibaGeek/ctfr>.

CTFR will read the public Certificate Transparency logs to 1) identify the owner SSL certificate in use at www.facebook.com and 2) see what other subdomains are also using the same SSL certificate.

Using CTFR to find related subdomains is very straightforward. Just execute the Python file and add your target domain. For our example, we are going to look for subdomains of Facebook.com:

```
root@OSINT: python ctfr.py -d facebook.com
```

```

  _ _ _ _ _
 / _ _ _ \
| | | | |
| | | | |
 \ _ _ _ /
  _ _ _ _ _

```

```
Version 1.2 - Hey don't miss AXFR!
Made by Sheila A. Berta (UnaPibaGeek)
```

```
[!] ---- TARGET: facebook.com ---- [!]
```

```

[-] *.adtools.facebook.com
[-] *.ak.facebook.com
[-] *.alpha.facebook.com
[-] *.assistant.facebook.com
[-] *.beta.facebook.com
[-] *.channel.facebook.com
[-] *.cinyour.facebook.com
[-] *.connect.facebook.com
[-] *.cstools.facebook.com
[-] *.ctscan.facebook.com
[-] *.dev.facebook.com
[-] *.dns.facebook.com
[-] *.extern.facebook.com
[-] *.extools.facebook.com
[-] *.facebook.com
[-] *.fb.alpha.facebook.com
[-] *.fb.beta.facebook.com
[-] *.fb.m.alpha.facebook.com
[-] *.fb.m.beta.facebook.com

```

```
[remaining 200+ subdomains truncated]
```

```
[!] Done. Have a nice day! ;).
```

With one simple scan, we have identified over 200 subdomains related to Facebook.com.

Easy, right? Since this is just the first step in an OSINT process, let's save the results of our scan to a file:

```
root@OSINT: python ctfr.py -d facebook.com -o subdomains.txt
```

Using certificate transparency gives us a new way of searching for subdomains without bruteforcing or even crawling around the site. This is completely stealth. What we did was look at the domain's certificate transparency logs, which tell us the other subdomains using the same shared SSL certificate.

Cool, right? Now let's apply that to more than just subdomains.

Crt.sh

Developed by Sectigo, crt.sh (www.crt.sh) is a certificate transparency log search engine. As new SSL certificates are issued, the certificate transparency logs will eventually (usually within a few hours) make their way to the crt.sh search engine. Not only can you search for active certificates, but crt.sh also allows you to see inactive, revoked, or expired certificates.

The search function is incredibly simple. Just put your search term in the search box. You can search for domain name, common name, organization name, or SHA hash.

To look at the certificate logs for my personal website, just type in my domain name and search (Figure 10.1).

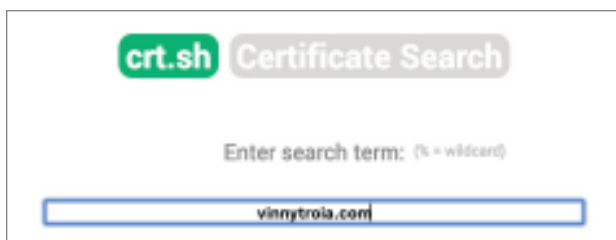


Figure 10.1

The results of your search will look something like Figure 10.2.

Each crt.sh ID represents a point-in-time snapshot of the certificate in use on the target website. Clicking a crt.sh ID will expand the details of a particular certificate, as shown in Figure 10.3.

CT in Action: Side-stepping Cloudflare

What if historical WHOIS data doesn't get you anywhere and you are looking for a few additional clues as to who the owner of the domain might be?

cert.sh ID	Issued At	Not Before	Not After	Issued By
1173687076	2019-02-03	2019-02-03	2019-05-04	C=US, O=Let's Encrypt, CN=Let's Encrypt Authority X3
1167310269	2019-02-03	2019-02-03	2019-05-04	C=US, O=Let's Encrypt, CN=Let's Encrypt Authority X3
1004601604	2018-12-05	2018-12-05	2019-03-05	C=US, O=Let's Encrypt, CN=Let's Encrypt Authority X3
1004601033	2018-12-05	2018-12-05	2019-03-05	C=US, O=Let's Encrypt, CN=Let's Encrypt Authority X3
830463037	2018-10-06	2018-10-06	2019-01-04	C=US, O=Let's Encrypt, CN=Let's Encrypt Authority X3
830538627	2018-10-06	2018-10-06	2019-01-04	C=US, O=Let's Encrypt, CN=Let's Encrypt Authority X3
830218563	2018-10-06	2018-10-06	2019-01-04	C=US, O=Let's Encrypt, CN=Let's Encrypt Authority X3
830218452	2018-10-06	2018-10-06	2019-01-04	C=US, O=Let's Encrypt, CN=Let's Encrypt Authority X3
830210025	2018-10-06	2018-10-06	2019-01-04	C=US, O=Let's Encrypt, CN=Let's Encrypt Authority X3
830249334	2018-10-06	2018-10-06	2019-01-04	C=US, O=Let's Encrypt, CN=Let's Encrypt Authority X3
826901964	2018-10-06	2018-10-06	2019-01-04	C=US, O=Let's Encrypt, CN=Let's Encrypt Authority X3
826976279	2018-10-06	2018-10-06	2019-01-04	C=US, O=Let's Encrypt, CN=Let's Encrypt Authority X3
11762189	2015-12-31	2015-12-30	2016-07-03	C=GB, ST=Greater Manchester, L=Salford, O=COM/ODD C.A.L
10901189	2015-11-26	2015-11-24	2016-05-07	C=GB, ST=Greater Manchester, L=Salford, O=COM/ODD C.A.L
10704875	2015-11-15	2015-11-13	2016-05-07	C=GB, ST=Greater Manchester, L=Salford, O=COM/ODD C.A.L
10418131	2015-10-31	2015-10-29	2016-04-24	C=GB, ST=Greater Manchester, L=Salford, O=COM/ODD C.A.L
10399742	2015-10-30	2015-10-29	2016-04-24	C=GB, ST=Greater Manchester, L=Salford, O=COM/ODD C.A.L
10237720	2015-10-20	2015-10-18	2015-12-30	C=GB, ST=Greater Manchester, L=Salford, O=COM/ODD C.A.L

Figure 10.2

cert.sh ID	10901189
Issued At	2015-11-26
Not Before	2015-11-24
Not After	2016-05-07
Issued By	C=GB, ST=Greater Manchester, L=Salford, O=COM/ODD C.A.L
Subject	CN=www.10901189.com
Public Key Algorithm	rsaEncryption
Public Key Size	2048
Signature Algorithm	sha256WithRSAEncryption
Signature Size	256
Signature	04 9a 77 06 0e 78 12 3d 44 42 75 78 6a d3 05 10

Figure 10.3

For this example, let's revisit our friend WhitePacket. The domain `www.whitepacket.com` has privacy enabled starting in 2015 (which is presumably when the current domain owner took ownership of the domain). In this case the historical WHOIS data is no help, but in looking at the data we can see that the domain's DNS is using Cloudflare.

Cloudflare's DNS provides a mechanism to mask a server's true IP address, making it extremely difficult to investigate. One thing I also love about Cloudflare is that it provides a shared SSL certificate that you can use for *all* domains in your account.

See where I'm going with this?

Using `www.crt.sh`, let's investigate `WhitePacket.com`'s Certificate Transparency logs, shown in Figure 10.4.

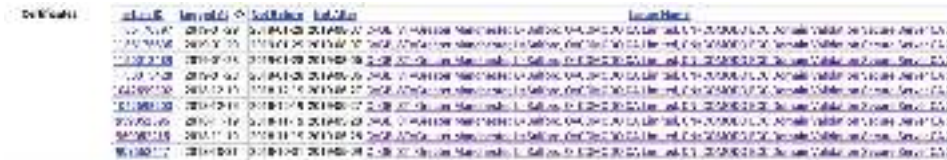


Figure 10.4

Clicking one of the `crt.sh` IDs will give you the same info we saw earlier. This time, scroll down to the “Subject Alternative Name” section and you can see the difference:

```
Certificate:
  Data:
    Version: 3 (0x2)
    Serial Number:
      83:23:1a:8d:16:53:53:8d:5f:73:a9:92:a0:3e:bf:8c
    Signature Algorithm: ecdsa-with-SHA256
    Issuer: (CA ID: 1582)
      commonName          = COMODO ECC Domain Validation
      Secure Server CA 2
      organizationName     = COMODO CA Limited
      localityName         = Salford
      stateOrProvinceName  = Greater Manchester
      countryName          = GB
```

[results truncated]

```
Authority Information Access:
CA Issuers - URI:http://crt.comodoca4.com/
COMODOECCDomainValidationSecureServerCA2.crt
```

```
X509v3 Subject Alternative Name:
  DNS:sni230260.cloudflaressl.com
  DNS:*.4origines.com
  DNS:*.agileriosegueros.com.br
```

```
DNS:*.ardyssv.gq
DNS:*.ayrescorretora.com.br
DNS:*.bleudecode.com
DNS:*.bolnichnye-listi.com
DNS:*.booksport1.cf
DNS:*.cabinetpaintingrefinishing.com
DNS:*.colombia-sexo.tk
DNS:*.donji-vakuf.tk
DNS:*.ebooksfree911.cf
DNS:*.galileiaseguros.com.br
DNS:*.girlsmakeup.tk
DNS:*.guessmantra.ga
DNS:*.homespaces.com
DNS:*.instapics.bid
DNS:*.instapics.party
DNS:*.jand.tk
DNS:*.meettabernacle.com
DNS:*.memoriaseguros.com.br
DNS:*.mindbendingbeats.ca
DNS:*.naysathseguros.com.br
DNS:*.og.money
DNS:*.originsgranite.com
DNS:*.palwalonline.in
DNS:*.peterkomorowski.com
DNS:*.phpturtle.com
DNS:*.reenuu.com
DNS:*.satanism.ca
DNS:*.seguros6k.com.br
DNS:*.spravka-moscow177.ru
DNS:*.tagbook-s.cf
DNS:*.toolzdepot.com
DNS:*.whitepacket.com
```

We have a literal slew of domains that are sharing the same SSL certificate as `WhitePacket.com`. In a *best-case scenario*, this will reveal domains that are all part of the same Cloudflare account. This does not necessarily imply they all of the same owner, just that they are all sharing the same Cloudflare account.

NOTE FROM Cloudflare

I reached out to someone on Cloudflare's investigative team regarding the use of certificates to find related domain owners. This was their response: "In the vast majority of cases completely unrelated domains end up grouped together on shared TLS SNI certificates."

Noted! As with anything OSINT related, this is just one more technique you can use to gain more clues as you spiral down the rabbit hole of information. In our test case, the information turned out to be valid. However, it's important to note that Cloudflare has pointed out that this may not always be the case. Tread carefully.

Testing More Targets

To Cloudflare's point, maybe these domains are unrelated. I happen to know from conversations with the owner that at least some of these are accurate. So for the sake of thoroughness, let's try a few more domains.

For our next target, we can look up the Certificate Transparency logs for `exploit.in`, a popular Russian hacking forum that also uses Cloudflare. (See Figure 10.5.)

[illegible]

Figure 10.5

Looking specifically at crt.sh ID 11915608, we can see the following results:

Certificate:

Data:

Version: 3 (0x2)

Serial Number:

40:7f:98:9b:8d:ef:73:2f:81:4c:a1:ef:ff:dc:28:c3

Signature Algorithm: ecdsa-with-SHA256

Issuer: (CA ID: 1582)

```
commonName          = COMODO ECC Domain Validation Secure Server CA 2
```

```
organizationName      = COMODO CA Limited
```

```
localityName      = Salford
```

```
stateOrProvinceName    = Greater Manchester
```

```
countryName      = GB
```

Validity

Not Before: Oct 21 00:00:00 2015 GMT

Not After : Oct 20 23:59:59 2016 GMT

Subject:

```
commonName      = sni309783.cloudflaressl.com
```

```
organizationalUnitName      = PositiveSSL Multi-Domain
```

```
organizationalUnitName    = Domain Control Validated
```

[. . .]

X509v3 Subject Alternative Name:

DNS:sni309783.cloudflaressl.com

DNS:*exploit.in

DNS:exploit.in

They use Cloudflare and it would appear that no other domain is using the same shared SSL.

To Cloudflare's point, I can run this tool in several instances and find completely unrelated domains. My domain happens to be one of them.

When you look up `vinnytroia.com` in `crt.sh`, you get the following results:

```
X509v3 Subject Alternative Name:
    DNS:sni68313.cloudflaressl.com
    DNS:*.alwatanSPORT.ps
    DNS:*.alwatanvoice.com.ps
    DNS:*.alwatanvoice.net.ps
    DNS:*.alwatanvoice.org.ps
    DNS:*.hospitalshub.com
    DNS:*.jenanmag.com
    DNS:*.larhonda.review
    DNS:*.pincodehub.com
    DNS:*.ramallahnet.com
    DNS:*.thepointatpentagoncityapts.com
    DNS:*.vinnytroia.com
    DNS:*.wholiesmore.org
    DNS:alwatanSPORT.ps
    DNS:alwatanvoice.com.ps
    DNS:alwatanvoice.net.ps
    DNS:alwatanvoice.org.ps
    DNS:hospitalshub.com
    DNS:jenanmag.com
    DNS:larhonda.review
    DNS:pincodehub.com
    DNS:ramallahnet.com
    DNS:thepointatpentagoncityapts.com
    DNS:vinnytroia.com
    DNS:wholiesmore.org
```

As it turns out, Cloudflare's point is accurate. With the exception of my site, `vinnytroia.com`, none of these domains belong to me. So again, *if you are going to use this method to look for additional clues, please be aware that the results may not always be accurate.*

CloudFlair (Script) and Censys

Cloudflare is intended to act as a middleman between a website and its users, in order to protect the website's owners from various types of cyber attacks. In theory, an attacker would not access a website directly. Instead, a user accesses a website through `Cloudflare.com`'s protective layer, which ultimately masks the server's true IP address. This can be a useful tool for security organizations to employ because it masks the company's original IP address, thereby making it more difficult to perform active recon against the site.

Of course, this is assuming the server is properly configured, which is not always the case. Or more importantly, maybe the server is properly configured *now*, but was not at an earlier date.

Similar to crt.sh, Censys.io can be used to search through certificate transparency logs to look for current or previously exposed servers.

How Does It Work?

Rather than looking for related domain names, we can also use the certificate searches to look for associated IP addresses. If these IPs are not part of the Cloudflare network, then we surmise that the IP belongs to our target domain.

If you would like to automate this process, there is a Python tool called CloudFlair (CF, so we don't confuse it with Cloudflare, the company). CF is a tool used to automate the process of finding the origin of publicly exposed servers that do not properly restrict access to the Cloudflare.com IP ranges.

This tool is available for download at <https://github.com/christophetd/CloudFlair>.

For our first example, let's see if we can use CF to track the originating server of Codepen.io:

```
root@OSINT: python cloudflair.py codepen.io

[*] Retrieving Cloudflare IP ranges from
https://www.cloudflare.com/ips-v4
[*] The target appears to be behind CloudFlare.
[*] Looking for certificates matching "codepen.io" using Censys
[*] 13 certificates matching "codepen.io" found.
[*] Looking for IPv4 hosts presenting these certificates...
[*] 3 IPv4 hosts presenting a certificate issued to "codepen.io"
were found.
  - 52.27.19.56
  - 54.201.58.131
  - 52.10.104.25

[*] Testing candidate origin servers
[*] Retrieving target homepage at https://codepen.io
[*] "https://codepen.io" redirected to "https://codepen.io/"
  - 52.27.19.56
  - 54.201.58.131
  - 52.10.104.25

[*] Found 1 likely origin servers of codepen.io!
  - 52.27.19.56 (HTML content is 96% structurally similar to codepen.io)
```

That's all there is to it!

Before I end this section, you know that thing I keep saying about not solely relying on one tool, and to make sure you test using multiple tools and techniques?

To further illustrate my point, look what happens when searching for WhitePacket.com on using the same tool:

```
root@OSINT: cloudflair.py whitepacket.com

[*] Retrieving Cloudflare IP ranges from
https://www.cloudflare.com/ips-v4
[*] The target appears to be behind CloudFlare.
[*] Looking for certificates matching "whitepacket.com" using Censys
[*] 0 certificates matching "whitepacket.com" found.

Exiting.
```

Zero certificates found, even though crt.sh shows multiple certificates for WhitePacket.com. Don't get me wrong—Censys.io is a paid tool, and a very good one. But no tool is perfect. Make sure you don't rely too heavily on one source or you might miss something.

Wayback Machine and Search Engine Archives

Wayback, or the Wayback Machine, is a digital archive of the web. Wayback (aka Archive.org) will allow you to see the state of a website, as it was, on a historical date.

This book has already covered reasons why looking at a website's historical state is an important way to search for clues. This section will mostly focus on ways to automate your searches, but first let's look at some basic examples.

As Figure 10.6 shows, when searching Wayback Machine for www.whitepacket.com, the results will show archived data between 2013 and 2018.



Figure 10.6

To further explore these archives, click a year, followed by a snapshot date. The blue circles on the calendar (see Figure 10.7) indicate the date a snapshot of a website was taken. To view an archive of the website created on a particular date, click any of the blue circles on the calendar.

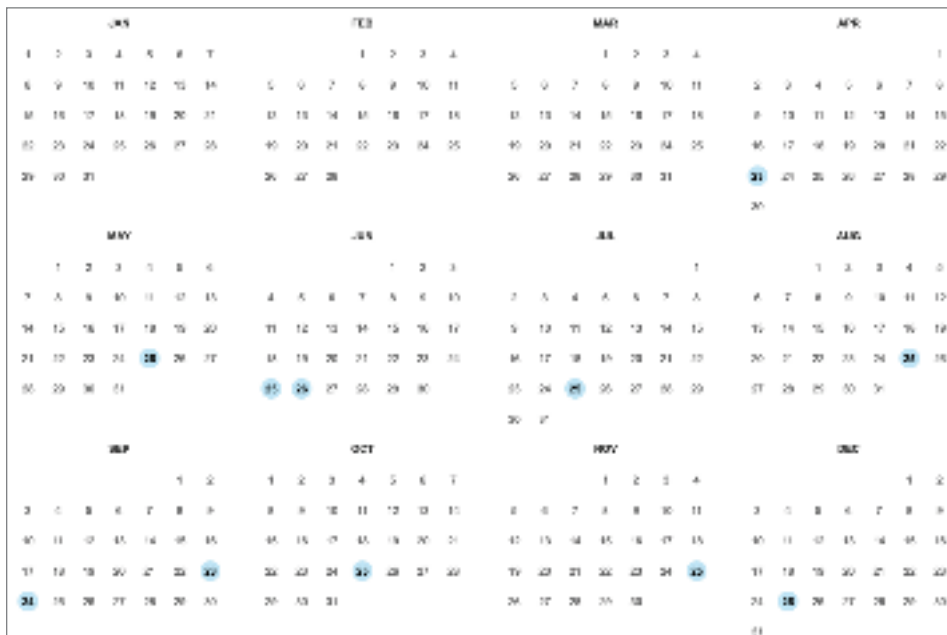


Figure 10.7

For most of 2017, WhitePacket forwarded his domain to a Twitter account: www.twitter.com/whitepacket. The Twitter account has been deleted, but sometimes we can get lucky using Google's cache.

Search Engine Caches

Sometimes Google will offer cached versions of websites through its standard search results. To access the cached versions of a site, click the green down arrow (▼) next to a search result and select "cached view." Let's try looking for a cached version of WhitePacket's Twitter page (shown in Figure 10.8) by entering the following search in Google:

```
site: twitter.com/whitepacket
```

Google only has one result that allows you to view a cached copy. Several tweets are listed, but clicking them brings us to a "page not found" error since the page was deleted.

Let's see if our results are different in Bing, shown in Figure 10.9.



Figure 10.8



Figure 10.9

The results with Bing are the same with respect to the cached copy of the Twitter page, but take a look at the last result—an open-source TOR botnet coded on WhitePacket's GitHub page. That looks interesting.

Exploring that page gives us an email address, jabber ID, and a BTC address to add to our list of search terms.

CachedView.com

CachedView.com (Figure 10.10) is a site that allows you to easily search for archived web pages across Google's cache, Archive.org, and Coral Cache. The results of the site are the same; it just provides a type of hub for each access of multiple cache sites.



Figure 10.10

Wayback Machine Scraper

The Wayback Machine (www.archive.org) can be an amazing source of information. When dealing with large sites, the amount of information you have to sift through can be overwhelming.

The best way I've found to do this is to literally scrape all cached copies of a site, then sift through the data locally. A few different tools are available that you can use to scrape the data. One of them, `enum_wayback`, is even baked right into Metasploit, thanks to Rob Fuller (aka Mubix).

EXPERTTIP: ROB FULLER

I suck at pen testing web applications. I'm not very good at figuring out how they're filtered and what they do. And since everything's a web application these days, that doesn't lend very well to breaking into them. So, knowing how an application works on the backend and knowing which websites or which URIs to go and target is sort of a guessing game.

So you have to kind of figure out where you need to log in to get to the slash admin panel, dashboard, etc. All of those URIs that go into the dashboard has a kind of

inherent vulnerability in it. Unless they code it right, you can directly access resources from the dashboard or panel.

The Wayback Machine records websites at different states. If the Wayback Machine was able to record a URI that is no longer in use or was open for a minute when it caught it, that can be really beneficial to knowing how an application works, or at least even just a history of it.

The `enum _ wayback` script was designed to kind of give you a bunch of URIs that you might not have known about. It can show you URIs that existed and may not exist now. The script won't copy all of the functionality of a web application, but it does spider it.

Let's say they had a custom version of Drupal one year and a custom version of WordPress the next year. You're going to get both Drupal and WordPress URLs or URIs. There's a good chance that Java either left files around or didn't set up the routing correctly, or some other detail, and there's going to be technical debt there. Then Wayback not only finds that, but it also finds anything that was open that probably shouldn't have been.

Let's use Bob's Web Store as an example. There's a good chance that if Bob's Web Store has been around for 15 years, they didn't think about security until 10 years ago, and so those URIs that were indexed and cataloged might still be there and still be very valid, but now more secure.

It might be that 80% of them are more secure, but you might just find that *one* that isn't—and that can really make the difference. The script is designed to give you a bigger resource of pages to look at.

Using it is really straightforward. All you do is point it at a specific host name or domain and it gives you the results from the Wayback, all parsed and nice and neat, so you don't have to page through anything and that makes it very easy to load into Burp, too.

Enum Wayback

`enum _ wayback` is a Ruby script that is now part of the Metasploit framework. Since this is a book on investigations and OSINT, we really won't be digging too far into Metasploit or all of the amazing things you can do with it. If you haven't used Metasploit in the past (or maybe don't know what it is), Metasploit will be one of your primary tools when conducting a penetration test. Hundreds (if not thousands) of exploits and customizable modules are built into the tool, making it a necessity for any pen-tester.

For those of you versed in Metasploit, we can launch `enum _ wayback` like this:

```
msf > use auxiliary/scanner/http/enum_wayback
```

To see our list of available options, we can type in options:

```
msf auxiliary(scanner/http/enum_wayback) > options
```

Module options (auxiliary/scanner/http/enum_wayback):

Name	Current Setting	Required	Description
----	-----	-----	-----
DOMAIN		yes	Domain to request URLs for
OUTFILE		no	Where to output the list for use

The options for the Wayback scraper are pretty straightforward. You can set the domain you wish to target, and the output file for the data.

Scraping Wayback with Photon

Many other command-line-based tools can be used to scrape Wayback archives. One of them is Photon, a tool we discussed in a previous chapter. The Photon Wayback scraper will fetch all of the available URLs of a site as “seeds.” In this context, a seed file is basically a list of other URLs that you can then go back and visit/scrape. In other words, Photon is acting like a spider and will crawl the different URLs available on Wayback and give you a nicely formatted list.

There will be many cases where websites (or forums) do not require authentication to see their posts. For whatever reasons, a number of hacker forums can be viewed without the need to log in, which means their content will be scraped and cached. One such website is Bezlica.top, an old Russian hacking forum.

I am using this forum specifically because it was one of the forums where TDO first posted the sale of its medical data. Figure 10.11 shows a screenshot of the original post.



Figure 10.11

Being able to index and historically catalog this data can be incredibly useful for future investigations. What if we wanted to search for more posts by this user? This happens to be a perfect example because the site is no longer online. Lucky for us, the site's posts were public and scraped by Wayback, which means we can grab them with Photon.

To run Photon's Wayback engine, use the `--wayback` switch.

When running Photon, we can also change the depth of the scan. This is useful if we know the site has many levels of nested pages, like forums:

```
python photon --wayback -d 10 -u bezlica.top
```

```
root@INTEL:/opt/Photon: python photon.py --wayback -u bezlica.top -d 10
```

```

  / _ \ / _ \ / _ \ / _ \ / _ \
 / _ \ / _ \ / _ \ / _ \ / _ \
 / _ \ / _ \ / _ \ / _ \ / _ \
 / _ \ / _ \ / _ \ / _ \ / _ \ v1.2.1

```

```

[~] Fetching URLs from archive.org
[+] Retrieved -1 URLs from archive.org
[+] URLs retrieved from robots.txt: 1
[~] Level 1: 2 URLs
[!] Progress: 2/2
[~] Level 2: 3 URLs
[!] Progress: 3/3
[~] Crawling 4 JavaScript files
[!] Progress: 4/4
-----
[+] Intel: 3
[+] Robots: 1
[+] Internal: 5
[+] Scripts: 4
[+] External: 3
-----
[!] Total requests made: 9
[!] Total time taken: 0 minutes 52 seconds
[!] Requests per second: 0
[+] Results saved in bezlica.top directory

```

Archive.org Site Search URLs

Many scraper tools may require you to provide them with a hard list of URLs to scrape. Building a list of working URLs can prove to be a very big challenge, especially when you are talking about forums with millions of posts. As it turns out, you can use specially crafted URLs to download lists of pages cached by Archive.org.

The following URL will give a list of each cached copy of a site, and its date, substituting the actual domain: <https://web.archive.org/cdx/search?url=domain.com>.

For example, entering this URL in your browser to search Bezlica.top: <https://web.archive.org/cdx/search?url=bezlica.top> gives us the following results:

```
top,bezlica)/ 20160911021317 http://bezlica.top:80/
text/html 200 7TMNRLBP5VIJEXO7K6RCZXPJTULOV5PO 11844
top,bezlica)/ 20160914070721 http://bezlica.top
text/html 200 OHXJUAIWGOH66WENAYCIG45PRYCHCCE 12492
top,bezlica)/ 20161016030541 http://bezlica.top:80/
text/html 200 26YUBAWKSXVQJYXTHOBOLCPOTCJ3PH5C 10878
top,bezlica)/ 20161208235225 http://www.bezlica.top:80/
text/html 200 BU2WMW7ZLZNL5VC6H6RLZVJCAEQUK2PH 11276
top,bezlica)/ 20161216210357 http://bezlica.top:80/
text/html 200 6GJ5JTHKFEGGJEGHK35V7XUQ4BALB4RC 10166
top,bezlica)/ 20170428055547 http://bezlica.top:80/
text/html 200 IJ6FUIV6CDPKDFOJDSFKWRVS37ORQHQ3 18429
top,bezlica)/ 20170509213741 http://bezlica.top:80/
text/html 200 55CY2ONBUCCJTKCOQH4DR2EOL2Z3OJA5 23358
top,bezlica)/ 20170520210324 http://bezlica.top:80/
text/html 200 7E3KPSJSOESAMJIX73Q6QGBOD3DFT734 21599
top,bezlica)/ 20170609030411 http://bezlica.top:80/
text/html 200 4QB2Y33ZP5HMM53EEAFSUV7YZO6AOX74 23576
top,bezlica)/ 20170611034742 http://bezlica.top:80/
text/html 200 JJE3IVA44G7P5YHIVR26HWSUS6BZDV7M 20182
top,bezlica)/ 20170630193537 http://bezlica.top:80/
text/html 200 NITYMLKCZ5XFDAPJWDCTHIQL2CJLOLIU 20882
top,bezlica)/ 20170712011807 http://bezlica.top:80/
text/html 200 4UBXAEQ2AWT5CL4USMJQXLWQEDOZI357 21177
top,bezlica)/ 20170724230545 http://bezlica.top:80/
text/html 200 PTPCXTUMBAUGRUKAGNXU7FDHRRKCKPKW 20240
top,bezlica)/ 20180104141200 http://bezlica.top:80/
text/html 200 6KQI5ILW4WE2EXH54ZRI6DWBNN5TR2H6 19138
top,bezlica)/ 20180307003146 http://bezlica.top:80/
text/html 200 7DJMCBRJBRLPMEZRLKFETZVPTCORNQBZ 18959
top,bezlica)/ 20180508100733 http://bezlica.top:80/
text/html 200 6POLETZBHK23AA7RRYRLHZUD3VPGAAX2 19171
top,bezlica)/ 20180609005028 http://bezlica.top:80/
text/html 200 UVFKHSNQK5R4OIUDZFL3XGYXB7WJXZST 19178
top,bezlica)/ 20180706234035 http://bezlica.top:80/
text/html 301 R4P57LSX7D3PMJ4CTPAJ3FU3VPVFIVF5 360
top,bezlica)/ 20180811090133 http://bezlica.top
text/html 301 Q4XK7WZCBMFO7HSN7SMGOAX7KLKFT6MT 497
top,bezlica)/ 20180904112205 http://bezlica.top
warc/revisit - Q4XK7WZCBMFO7HSN7SMGOAX7KLKFT6MT 457
top,bezlica)/ 20181112133152 http://bezlica.top
warc/revisit - Q4XK7WZCBMFO7HSN7SMGOAX7KLKFT6MT 458
top,bezlica)/ 20181127220905 https://bezlica.top/
warc/revisit - Q4XK7WZCBMFO7HSN7SMGOAX7KLKFT6MT 455
```

At the beginning of each line, we see the URL (top, bezlica) followed by the date and time of each crawl (2018-11-27, etc).

This information can then be saved and fed into a webscraper (like Photon) to go out and scrape each copy of the site to ensure we are collecting as much historical information as possible.

Wayback Site Digest: A List of Every Site URL Cached by Wayback

We can expand further our URL query to show every site-specific URL cached by Wayback by adding `&collapse=digest` to our URL string:

```
https://web.archive.org/cdx/search?url=domain.com*&collapse=digest
```

By looking at this URL, we can see a full list of every page cached for a domain.

Expanding our search to look at Bezlica.top again, we can see hundreds of pages of results that look like this:

```
https://web.archive.org/cdx/search?url=bezlica.top*&collapse=digest
top,bezlica)/threads/nuzhen-ship-telefona.30788/reply?quote=182018
20170504095614 http://bezlica.top:80/threads/nuzhen-ship-telefona.30788/
reply?quote=182018 text/html 200 6ULJU6ZH7ZM2ZGG6HNWHSDFWXC7QAAU 9611
top,bezlica)/threads/nuzhen-ship-telefona.30788/reply?quote=195845
20170504101949 http://bezlica.top:80/threads/nuzhen-ship-telefona.30788/
reply?quote=195845 text/html 200 AAKCFAPVFPA6HVSUY2S72XTK6DPBWD4U 9639
top,bezlica)/threads/nuzhen-ship-telefona.30788/reply?quote=
196094 20170425161906
top,bezlica)/threads/nuzhna-rabota.12142 20170425002556
http://www.bezlica.top:80/threads/nuzhna-rabota.12142
text/html 301 3I42H3S6NNFQ2MSVX7XZKYAYSCX5QBYJ 428
top,bezlica)/threads/nuzhna-rabota.12142 20170425002556
http://www.bezlica.top:80/threads/nuzhna-rabota.12142/
text/html 200 75PDP3FGKT5AYHH5CT3WHF3DWW5GR43 11297
top,bezlica)/threads/nuzhno-fake-id.62868 20170425020755
http://bezlica.top:80/threads/nuzhno-fake-id.62868
text/html 301 3I42H3S6NNFQ2MSVX7XZKYAYSCX5QBYJ 424
top,bezlica)/threads/nuzhno-fake-id.62868 20170425020755
http://bezlica.top:80/threads/nuzhno-fake-id.62868/
text/html 200 73Y7JISNW7AXJDXTEB2GRCKC7OUGKPQO 8630
top,bezlica)/threads/nuzhny-sellery.90646 20170710104915
http://bezlica.top:80/threads/nuzhny-sellery.90646
text/html 301 3I42H3S6NNFQ2MSVX7XZKYAYSCX5QBYJ 405
top,bezlica)/threads/nuzhny-sellery.90646 20170710104915
http://bezlica.top:80/threads/nuzhny-sellery.90646/
text/html 200 4UTAHZPB5YVNTS5MSKJJ2JJ4FXACULRU 9200
top,bezlica)/threads/nuzhny-sellery.90646 20170712001847
http://bezlica.top:80/threads/nuzhny-sellery.90646/
text/html 200 PIVOCNJCPNWRDOC2XFHVKUZPNNDNI4EG 9204
top,bezlica)/threads/obnal-vcc.22482 20170425023630
http://bezlica.top:80/threads/obnal-vcc.22482
```

```
text/html 301 3I42H3S6NNFQ2MSVX7XZKYAYSCX5QBYJ 423
top,bezlica)/threads/obnalichu-vashi-zvonki-s-chego-ugodno.
90587 20170722053208 http://bezlica.top:80/threads/obnalichu-
vashi-zvonki-s-chego-ugodno.90587 text/html 301
3I42H3S6NNFQ2MSVX7XZKYAYSCX5QBYJ 547
top,bezlica)/threads/obnalichu-vashi-zvonki-s-chego-ugodno.90587
20170722053208 http://bezlica.top:80/threads/obnalichu-vashi-zvonki-
-s-chego-ugodno.90587/ text/html 200 DHB3ECDNAZZJTZKTSZ4X6C7U4MKIWQMN
10902
top,bezlica)/threads/obnalichu-vashi-zvonki-s-chego-ugodno.90587
20170723200538 http://bezlica.top:80/threads/obnalichu-vashi-zvonki-
-s-chego-ugodno.90587/ text/html 200 MB77HKZM7MH5YENQLZLLH4LKWF36TA5Q
11025
```

It's a bit difficult to read in one giant blob, but looking at the results we can see the full page URL that was cached, as well as the date the page was cached.

Why is this important? Rather than having to download all of the cached pages and look through a potential mountain of data, how easy do you think it would be to find an admin panel by searching through a list of URLs?

Now that we have the entire sitemap of cached pages, we can download a list of all cached pages and quickly search for specific keywords like “admin,” “panel,” and so on.

Remember, if someone wants to hide a URL from a search engine, all they need to do is update the `robots.txt` file on their website. Just because a site's current sitemap does not list an admin panel, that does not mean the panel did not exist at one point. It's possible the page is still there, just hidden or renamed. Using this method allows you to quickly discover pages that may have existed at one point but have since been removed.

Summary

This chapter focused on techniques that you can use to further investigate websites and domain ownership using Certificate Transparency and Internet archives (such as search engine caches and the Wayback Machine). Certificate Transparency provides a way to potentially find related websites (and website owners) by looking for shared SSL certificates. Cached search pages and Internet archives such as the Wayback Machine can be used to dig up interesting data on a site by looking at the history of its pages.

In the next chapter, we will focus on Iris by DomainTools, which has been the single most important tool in any of my investigations requiring information on domain history and historical website research.

Iris by DomainTools

If I had to pick a single tool that was most useful while investigating The Dark Overlord, it would be Iris.

DomainTools Iris is easily the most comprehensive and impressive historical domain registration search tool on the market. It is a full threat intelligence and investigation platform focused on providing context on threats with domain registration and Passive DNS data.

Iris is a paid proprietary tool. I do my best to include tools that are open source, but the truth is I have not found any other tool that even comes close to the capabilities I will demonstrate.

DomainTools not only has the most comprehensive database of historical domain registration data I have ever seen, but the tool itself is pretty amazing. (No, they are not paying me to write this—but now that I think about it, I probably should have asked.)

The Basics of Iris

This section will only cover the basics of Iris as we explore my personal website. We will dive deeper into Iris in subsequent sections as we start to combine it with other techniques. Iris is a great tool by itself, so combining it with other tools and techniques will make it that much more powerful, especially once we

start pulling all of the discovered information together in our tracking matrix (which we will begin constructing in Part IV of this book).

We can start using Iris by searching for any number of fields, including domain names, personal names, email addresses, physical addresses, IP addresses, SSL hash, DNS servers, and any other information that makes up a WHOIS record. (See Figure 11.1.)



Figure 11.1

Using the Iris search bar, we will start by looking up my personal site, “vinnytroia.com”. The results of the search will be displayed in Iris’s Pivot Engine view, shown in Figure 11.2.



Figure 11.2

The Iris Pivot Engine is really the cornerstone of the investigation subject. As you search for new items, the Pivot Engine will highlight which other paths you can explore within the domain WHOIS data. Each path is referred to as a *pivot*. Every time you take a new step in a different direction, you are pivoting from your original step. The data displayed in the Pivot Engine table shown in Figure 11.2 is very wide—it has 20 columns, which can highlight any number of searchable fields, including (but not limited to):

- Email
- Domain
- Registrant

- Organization
- Status
- Create/Expiration date
- Google Analytics
- MX or SPF records
- SSL certificates hash, origins, or country

Guided Pivots

Guided Pivots are intended to help identify potentially valuable pivot points; thus, they are highlighted for you. By default, Iris highlights any pivot point that connects to less than 500 domains (the threshold can be configured). As we roll over the “3 Guided Pivots” link, Figure 11.3 shows which pivots are available for further exploration.

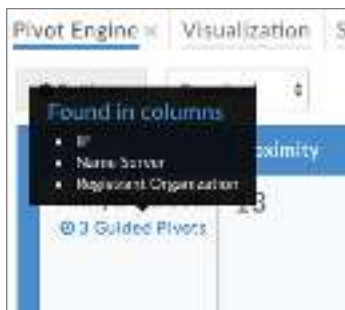


Figure 11.3

Configuring Your Settings

The Pivot Engine is fairly customizable. Clicking the Settings button opens the Settings panel. From here you can change which table headings are displayed in the Pivot Engine, as well as the ordering of the information being displayed in the Pivot Engine table.

In addition to the cosmetic items, you can also configure the sensitivity of your Guided Pivots (shown in Figure 11.4).

As discussed earlier, Iris highlights Guided Pivots that share values with less than 500 different domains (or matches). This number can be pretty high, so you can tone that down in this settings view. You can drop this number as low as 10, which means that the Guided Pivots highlight will only appear if the uniqueness of the data being displayed is found on no more than 10 different domains.

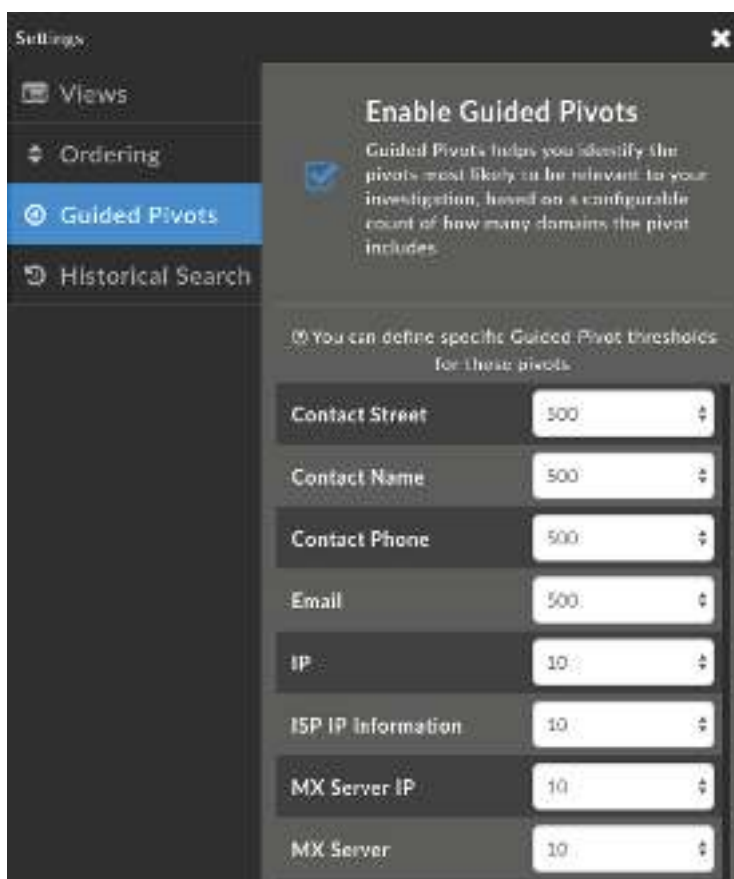


Figure 11.4

This can be a very useful setting when researching IP addresses because it can provide a quick way to isolate IPs that are not shared. For example, if you find a domain on an IP address that has only been used by a handful of other domains, there is a very good chance that all of the domains will be owned or operated by the same person.

Historical Search Setting

One last option is the Historical Search setting. By default, Iris searches are limited to matches where the search term appears in a current record. This means that if you searched for Vinny Troia, you would not find a domain that I registered five years ago that was sold to another owner. When searching, it is generally a good rule to make sure this setting is turned on unless there is a reason for not wanting older historical data in your results. (See Figure 11.5.)

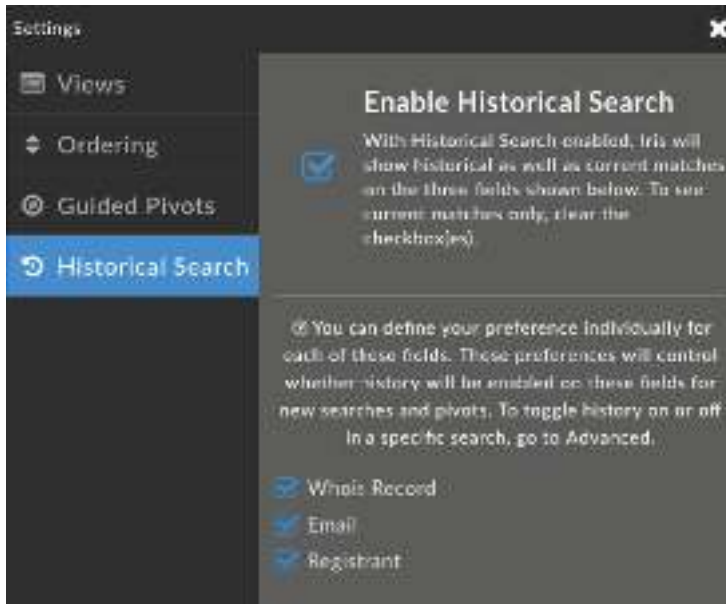


Figure 11.5

When Historical Search is enabled, the domain results will appear as either “active” or “inactive” in the results list, which is illustrated by a hyperlink icon for active domains, and a broken icon for inactive domains.

Pivootttt!!!

Every time I hear the word *pivot*, all I can think about is that one scene in *Friends* where Ross tried to move the couch up the stairs and he just kept yelling “Pivot!!!”

Moving on with our first pivot, we see in Figure 11.6 that the Pivot Engine has found three Guided Pivot points: IP, Name Server, and Registrant Organization.

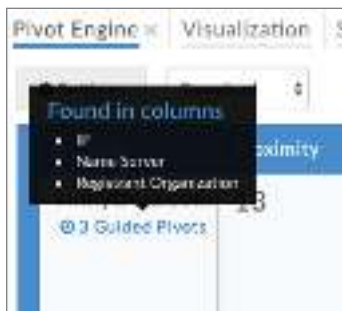


Figure 11.6

Scrolling across the list of columns in the Pivot Engine, the IP address is highlighted as a potential pivot point, as shown in Figure 11.7.

Name Server		IP			
Hostname	IP Information	IP	ISP/ IP Information	ASN	Country Code
ns1.curve.net	208.94.148.4	192.241.180.214	DigitalOcean LLC	14061	US
ns2.curve.net	208.80.126.4				
ns3.curve.net	208.80.126.4				

Figure 11.7

This seems like a good place to start our journey down this rabbit hole. To expand our search into this IP address, right-click the IP and select Expand Search (Figure 11.8).

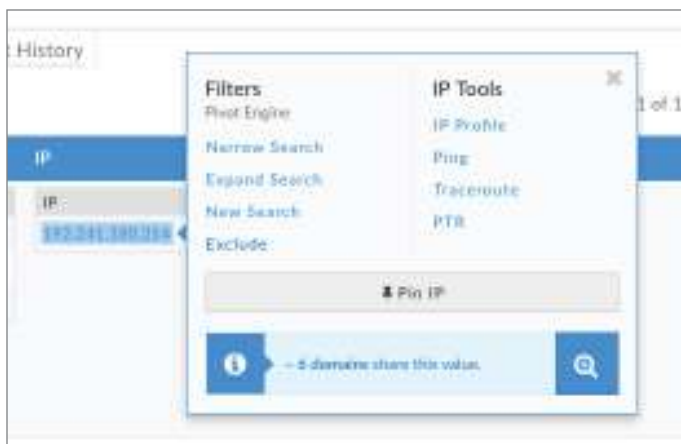


Figure 11.8

The UI data within the Pivot Engine will change to now showcase any domains using (or that have used) our current IP address search scope (Figure 11.9).

Looking at the results, you see a number of domains in the result set, all of which are owned by me. No great surprise here, but as we start applying this approach to identifying domains owned by actual threat actors, the results will appear much more interesting.

Keep in mind that finding domains that are hosted on unique (nonshared) IP addresses can be a rare occurrence, so let's continue our search by looking for other pivotable fields.

Domain	Hosts	Drill
countmeupthings.com [Go to Domain]	15	- Address 150.135.100.100 150.135.100.101 150.135.100.102 150.135.100.103 150.135.100.104 150.135.100.105 150.135.100.106 150.135.100.107 150.135.100.108 150.135.100.109 150.135.100.110 150.135.100.111 150.135.100.112 150.135.100.113
nightlionsec.com [Go to Domain]	15	- Address 150.135.100.100 150.135.100.101 150.135.100.102 150.135.100.103 150.135.100.104 150.135.100.105 150.135.100.106 150.135.100.107 150.135.100.108 150.135.100.109 150.135.100.110 150.135.100.111 150.135.100.112 150.135.100.113
nightlionsecurity.com [Go to Domain]	10	- Address 150.135.100.100 150.135.100.101 150.135.100.102 150.135.100.103 150.135.100.104 150.135.100.105 150.135.100.106 150.135.100.107 150.135.100.108 150.135.100.109

Figure 11.9

Pivoting on SSL Certificate Hashes

SSL certificates are a great way to categorize domains and will often provide connections between related infrastructure. The previous chapter showed the power of being able to find related domains using SSL certificate hashes or subject names. Iris brings that functionality into its own platform, allowing investigators to follow leads based on the same SSL certificate transparency information.

Looking at the certificate information for my personal domain, VinnyTroia.com, and my company domain, NightLionSecurity.com (shown in Figure 11.10), you see obvious differences in the SSL certificate.

Two of the domains share a similar SSL hash and subject name. The NightLionSecurity.com hash and subject is unique to that domain, which is why the data is not highlighted. There are no other matches.

To explore one of the highlighted SSL hash pivot points, right-click the hash and select Expand Search to expand your search to also include domains using this new SSL hash (Figure 11.11).

You can also choose to start a new search, or narrow your search results, meaning only sites that share the selected SSL hash will be displayed. Until now, previous examples have used the Expand Search option, which continues to add to the overall number of matches.

SSL Information	
☐ Domain: ☐ curvve.net Inspect Get Screenshot	Hash: 8bc236d0237b6e0d7e5d9f999177a0e0b0c Subject: CN=www.curvve.net, OU=Domain Control Validation, O=www.curvve.net
☐ curvverecordings.com Inspect Get Screenshot	Hash: 836ad162ed16825e2111385711e45e2796943e Subject: emailAddress=@info@curvve.com, CN=Flux, OU=Flux, O=Curvve, ST=Mississippi
☐ highlights.net Inspect Get Screenshot	Hash: 8bc236d0237b6e0d7e5d9f999177a0e0b0c Subject: CN=www.curvve.net, OU=Domain Control Validation, O=www.curvve.net
☐ highlightssecurity.com Inspect Get Screenshot	Hash: 013e4d0012c70e0e32c59f948124120e0d3e Subject: CN=*.highlightssecurity.com, OU=PrivateSSL, WILDCARD, O=Domain Control Validation, O=www.curvve.net

Figure 11.10



Figure 11.11

Figure 11.12 shows an expanded search for the SSL hash on my private web server.

Now we're cooking! After searching for SSL certificates with a similar hash value, we can see at least three new sites that were not in the previous list.

Keeping Notes

Every time you discover a new piece of content, you should be adding it to a mindmap, Excel sheet, or some “thing” that you can go back and easily reference. The farther down the rabbit hole you travel, the more information you will uncover, and it won't take any time at all before you are forgetting and losing track of potentially key pieces of information. You need a way to document and track all of the things you find along the way (I provide details on my own method starting in Chapter 16).

Domain		IPV Information		SSL Information	
☐ danielleya.com Inspect 0 Trusts 0 Guided Paths	v=sp1 +s +no -all	Host ylx234u53T6ek6d7abw994WY77aue2eb	Subject CN=www.serve.net,OU=Domain C		
☐ nightlion.net Inspect 0 Guided Paths	v=sp1 +s +no -all	Host ylx234u53T6ek6d7abw994WY77aue2eb	Subject CN=www.serve.net,OU=Domain C		
☐ nightlionsecurity.com Inspect 5 Guided Paths	v=sp1 ip4.142.243.224.200 include:spf.protection.outlook.com include:_spf.google.com include:spf.mandrillapp.com include:server.mcy.net -all	Host 013c5e866d0f604c32c99948832473baed3e	Subject CN=nightlionsecurity.com		
☐ troiadesigns.com Inspect 7 Guided Paths	v=sp1 +s +no -all	Host ylx234u53T6ek6d7abw994WY77aue2eb	Subject CN=www.serve.net,OU=Domain C		

Figure 11.12

Iris has its own method for keeping track of your investigation trails. Iris provides a “notes” section within each pivot point where you can document what you were doing, why you chose to pivot in a particular direction, and any other notes that you want to leave yourself to help you remember why you ended up where you did. Figure 11.13 shows the notes window in Iris.

The screenshot shows the VirusShare web interface. At the top, there's a search bar with the query "curvedmedia.com" and a filter set to "All". Below the search bar, there's a visualization of the search results, showing a network graph with nodes and edges. The main content area displays a list of search results for "curvedmedia.com". The results are organized into columns: "Domain", "Email", and "Address". The first result is "curvedmedia.com" with an IP address of "172.17.0.1" and a status of "2,947". The second result is "curvedmedia.com" with an IP address of "172.17.0.1" and a status of "2,947". The third result is "curvedmedia.com" with an IP address of "172.17.0.1" and a status of "2,947".

Figure 11.13

WHOIS History

Following our exploration of the SSL certs and related domains, a next logical step is to look up the WHOIS history of all the new target domains. We already found a handful just by looking up similar registrant information, and in the previous section we found a few more by looking for shared SSL certificates.

The WHOIS History tab on Iris is the most complete record of domain registration available. Period. I have yet to find a source for registration data that goes as far back as DomainTools (which is nearly two decades). Figure 11.14 shows the WHOIS history records after running a search on my personal domain, vinnytroia.com.

The screenshot shows the WHOIS History tab for the domain vinnytroia.com. The interface includes a navigation bar with tabs: Profile, Registrations, Status, IP Lookup, **WHOIS History**, Hosting History, and Subdomains History. Below the navigation bar, there are two main sections: 'Historical Records' and 'Whois Record for 2019-01-12'.

Historical Records: This section displays a list of domain registration records, each with a date and a 'changes' link. The records range from 2019-01-12 to 2003-01-12.

Whois Record for 2019-01-12: This section provides detailed information for the selected record. It includes fields for Domain, Record Date, Registrar, Server, Creation, Updated, Expires, and View Changes. Below this, there is a section for 'Domain Details' which contains a detailed text-based WHOIS record for the domain vinnytroia.com, including registration information, contact details, and DNS records.

Figure 11.14

The WHOIS history on this tab is organized by date—with my domain, the results go as far back as 2003.

For the curious reader, you can look at my 2003 results and see the address to my old apartment in New York City. (See Figure 11.15.)

[Is](#)
[Whois History](#)
[Hosting History](#)
[Screenshot History](#)

vinnytroia.com

Whois Record for 2003-08-15

[Previous](#)

Domain	vinnytroia.com
Record Date	2003-08-15
Registrar	GO DADDY SOFTWARE, INC.
Server	whois.godaddy.com
Created	2002-01-11
Updated	
Expires	2004-01-11

Unique Emails

- info@curve.com

```

Registrant:
  Curve
  401 West 54th St
  Suite 3B
  New York, New York 10019
  United States

Registered through: GoDaddy.com
Domain Name: VINNYTROIA.COM
Created on: 11-Jan-02
Expires on: 11-Jan-04
Last Updated on: 13-Oct-02

Administrative Contact:
  Troia, Vinny info@curve.com
  Curve
  401 West 54th St
  Suite 3B
  New York, New York 10019
  United States
  (877) 379-0395 Fax -- (877) 379-0395

Technical Contact:
  Troia, Vinny info@curve.com
  Curve
  401 West 54th St
  Suite 3B
  New York, New York 10019
  United States
  (877) 379-0395 Fax -- (877) 379-0395

Domain servers in listed order:
  NS.CISCOPT.COM
  NS2.CISCOPT.COM
  NS3.PROPAGATION.NET
  
```

Figure 11.15

NOTE This is very powerful stuff, especially when you consider the fact that most threat actors were skidz at one point and their operational security (OPSEC) probably wasn't that strong. A skid, or script-kiddie, is a term used to describe young and inexperienced hackers.

It never fails: if you go back far enough, you will always find a point where they were too young and naive to realize they left a trail. Vanity is always a major theme for blackhats—they always want the world to know of their accomplishments. This is exactly what happened with threat actor Cyper (more about him in the next chapter).

Oftentimes, the actor's vanity will lead back to some misstep or misconfiguration, potentially in domain name registration, which is why Iris is one of the most powerful tools in my arsenal.

Screenshot History

I find this section of Iris particularly useful because it means I don't have to go combing through the Wayback archives for every potential change. Of course, DomainTools may have sometimes missed a screenshot or site change, so results should always be confirmed and validated through manual testing (in other words, go back and make sure you didn't miss anything).

Looking at the screenshot history for my site (Figure 11.16), we can see results going back to 2004. This is a great example because I'm fairly positive my first website launched earlier than that. This is a perfect case where Wayback will most likely have additional information that is not on Iris.

If there is still a question of why this is important, keep reading. The end of this chapter will show a live example of how a researcher can bring all of this information together to form a concrete conclusion.

Hosting History

Finally, the Hosting History tab in Iris can provide even more clues as to a domain owner based on where the IP is located. It should come as no surprise that threat actors can (and often do) point subdomains to their home IP address. Either way, knowing where a site is hosted can turn out to be an important piece of information. If a site is hosted on Amazon, Azure, or some other giant hosting provider, it is probably a dead end for that line of research unless you can link that with some other piece of data.

Figure 11.17 shows the Hosting History tab for my personal website, which includes the different pre- and post-action IP addresses. This can be extremely useful if someone decides to host a domain from a private server, such as their house.

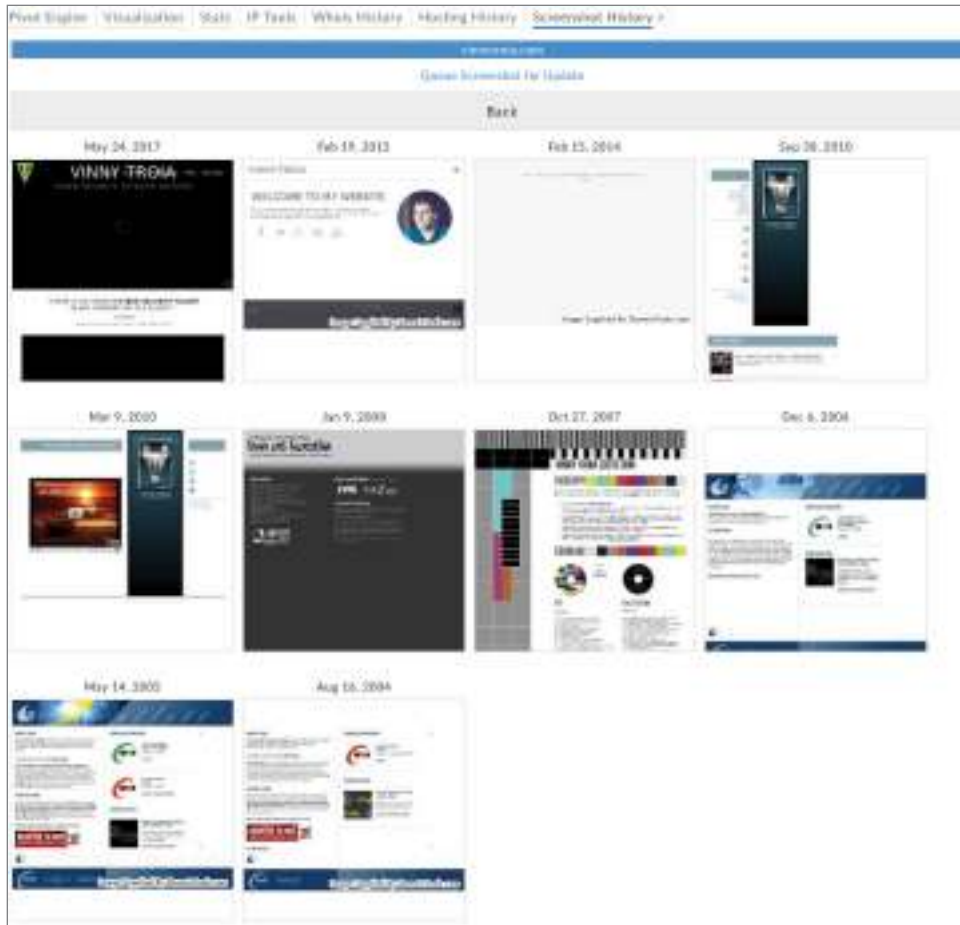


Figure 11.16

The uncovered hosting history shows all of the different IPs from my web server going back to 2003. One possible search at this point is to look into each of the IPs to see if any other malicious or questionable sites are tied to those IPs.

This will be a very time-consuming process, so one suggestion would be to use Iris's notes feature mentioned earlier in this chapter. Iris's notes can be placed on any pivot point and search you perform, which will make for a very good way to help you backtrack and understand how or why you came to a particular point in your research.

You never know when a piece of information (such as an IP) will come up later in an investigation, so make sure to document your results in a way that is easily searchable later.

Privat Engine	Visualization	Status	IP Tools	Whois History	Hosting History	Serverman History
WhoisHistory.com						
IP Address History	Register History	Name Server History				
First Date	Action	Pre-Action IP	Post-Action IP			
2008-11-05	Change	192.168.169.101	192.168.168.141			
2008-10-17	Change	162.242.121.100	192.168.169.101			
2006-01-03	Change	104.27.154.21	162.242.121.100			
2005-06-24	Change	104.28.4.134	104.27.154.21			
2005-02-08	Change	103.162.192.152	104.28.4.134			
2005-02-01	Change	103.168.4.121	100.160.160.177			
2004-11-05	Change	107.170.142.112	104.28.4.134			
2004-07-09	Change	162.242.167.42	107.170.142.112			
2004-05-31	Change	198.211.113.108	162.242.167.42			
2003-06-12	Change	107.23.116.26	198.211.113.108			
2002-11-01	Change	205.186.167.181	107.23.116.26			
2001-03-06	Change	74.206.238.129	205.186.167.181			
2001-04-10	New	none	74.206.238.129			
2001-02-13	Change	74.206.238.124	74.206.238.129			
2000-12-19	Change	74.206.109.35	74.206.238.124			
2000-06-29	Change	74.206.45.22	74.206.109.35			
2000-05-04	Change	69.132.145.247	74.206.45.22			
2000-03-01	New	none	69.132.145.247			

Figure 11.17

Iris's notes feature will help you keep track of every search and pivot you make. The notes are also exportable, which is amazing in itself. When you are done, you can export all of your notes in one giant file for reference later.

Bringing It All Together

The importance of scraping older versions of a website was discussed earlier in this chapter, but to drive the point home, let's take another quick look at our friend WhitePacket.

In Chapter 9, you saw that the domain registration data for `WhitePacket.com` showed a lapse in owner information (i.e., the domain expired) between 2014 and 2015. Figure 11.18 shows the domain registration history for `WhitePacket.com`. (Whenever the domain registration data is private, there will be a little “eye” icon to let you know: .)

We can't assume the owners to be the same person. In fact, let's go with the assumption they are not. The first thing we might do is look at the historical domain data to see if we can find any interesting clues about the domain's new owner. Figure 11.19 shows historical screenshots of the domain in question.

2015-12-20	changes
2015-11-24	changes
2015-11-21	changes
2015-10-20	changes
2015-10-19	changes
2015-10-18	changes
2014-03-01	changes
2013-12-25	changes
2013-12-22	changes
2013-11-14	changes
2013-02-05	changes

Figure 11.18



Figure 11.19

Using Iris's Screenshot History tab, there is an obvious gap of time between 2016 (when the site looked like it redirects to Twitter) and 2013. Let's circle back and see what kinds of results we will find using the Wayback Machine at archive.org. Figure 11.20 shows the archive.org timeline feature discussed in Chapter 10.

What we can see from this screenshot is that archive.org shows 21 screen captures between July 20, 2013 and November 30, 2018. It is interesting to note that the results in Iris go further back, but Wayback has more data (just another friendly reminder never to rely on a single source for all of your information gathering).

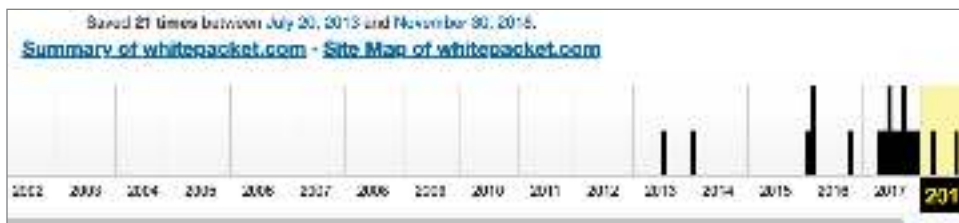


Figure 11.20

Digging through the page archives, there is nothing worth noting in 2015. Moving on, a review of the first capture in 2016 pulls up an archived page that gives us a plethora of new information, including the testimonial shown in Figure 11.21.

I rarely write testimonials, but Mr. Meunier (WhitePacket) is one of the exceptions. Some months ago we got a warning from an XSSposed report about a potential threat on one of our websites. With a big concern from upper management and IT security, we were trying to patch the hole as quick as we could. I was testing my luck and contacted WhitePacket directly as he was the original white-hat hacker who found the bug, surprisingly enough, he did respond! Thanks, WhitePacket (Christopher Meunier) for giving us the clear explanation and the valuable advice, we had the patch in a matter of just a few hours. I'm not in the position of representing my organization but from the bottom of my heart, I again thank Mr. Meunier for his hard work, rich knowledge, and the good heart of being a white-hat hacker.

Figure 11.21

In addition to the testimonial that states the owner's name, Figure 11.22 shows additional contact information. Bingo.



Figure 11.22

The contact information located on the page lists two phone numbers, a physical address, an email address (#4) that links to another domain, and three social media pages. We now have eight new pieces of information to explore.

IMPORTANT: DO NOT STOP HERE

Most people will take this information and immediately start following the new rabbit hole and digging into the new clues. This is where I would advise exercising some restraint. As exciting as it is to follow a fresh lead, doing so will most likely mean that you will forget to go back and finish searching through [Archive.org](https://archive.org) (or whatever it is that you are searching on at the moment). There can be more clues to find in your current investigative process (and in this case, there are). Just a note of caution: If you immediately leave to go exploring, important clues can (and will) be missed. In other words, try to hold back on the impulse to jump to something new and finish your current task first. Spending a little extra time now can potentially lead to a bigger payoff later.

As a case in point, future revisions of the site show an updated email address and phone number, as shown in Figure 11.23.

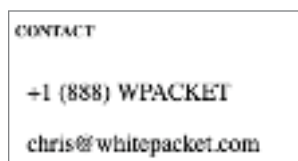


Figure 11.23

Once all possible site archives have been explored and exhausted, we can go back and start researching some of our new information. Earlier in this chapter, we showed that the SSL certificate information for [WhitePacket.com](https://whitepacket.com) was also associated with 46 other domains.

We can't say with any degree of certainty that all of the domains belong to the same owner, so the only thing we can do is check. Fast-forwarding through all of the manual research, we find two domains of interest:

- [MindBendingBeats.ca](https://mindbendingbeats.ca)
- [Anonimo.ninja](https://anonimo.ninja)

[MindBendingBeats.ca](https://mindbendingbeats.ca) is a site that sells binaural beats for Bitcoins. Maybe we can get lucky and find a crypto wallet address in the archives. To expedite this process, we can use Photon (discussed in Chapter 10) or another useful Python tool called [wayback-machine-scraper](https://github.com/wayback-machine/wayback-machine-scraper) to download an archive of all the pages cached by the Wayback Machine (archive.org).

The following code shows the output of using [wayback-machine-scraper](https://github.com/wayback-machine/wayback-machine-scraper) to accomplish this task:

```
2019-02-26 08:18:33 [scrapy.core.engine] INFO: Spider opened
2019-02-26 08:18:33 [scrapy.extensions.logstats] INFO: Crawled 0 pages
```



```
(at 0 pages/min), scraped 0 items (at 0 items/min)
2019-02-26 08:18:37 [scrapy.core.engine] INFO: Closing spider (finished)
2019-02-26 08:18:37 [scrapy.statscollectors] INFO: Dumping Scrapy stats:
{'downloader/request_bytes': 6021,
 'downloader/request_count': 17,
 'downloader/request_method_count/GET': 17,
 'downloader/response_bytes': 72542,
 'downloader/response_count': 17,
 'downloader/response_status_count/200': 17,
 'dupefilter/filtered': 32,
 'finish_reason': 'finished',
 'finish_time': datetime.datetime(2019, 2, 26, 8, 18, 37, 320617),
 'log_count/INFO': 7,
 'memusage/max': 50098176,
 'memusage/startup': 50098176,
 'offsite/domains': 8,
 'offsite/filtered': 40,
 'request_depth_max': 1,
 'response_received_count': 14,
 'scheduler/dequeued': 26,
 'scheduler/dequeued/memory': 26,
 'scheduler/enqueued': 26,
 'scheduler/enqueued/memory': 26,
 'start_time': datetime.datetime(2019, 2, 26, 8, 18, 33, 470619)}
2019-02-26 08:18:37 [scrapy.core.engine] INFO: Spider closed (finished)
```

Wayback-machine-scraper returned eight cached pages, and upon looking through the code, none of them contained a BTC wallet. Too bad. Also, the pages did not contain anything of value. Next, let's look at Anonimo.ninja.

Looking at the site on Archive.org, we can immediately see in Figure 11.24 that the owner is the same because of the associated Twitter page (@WhitePacket).

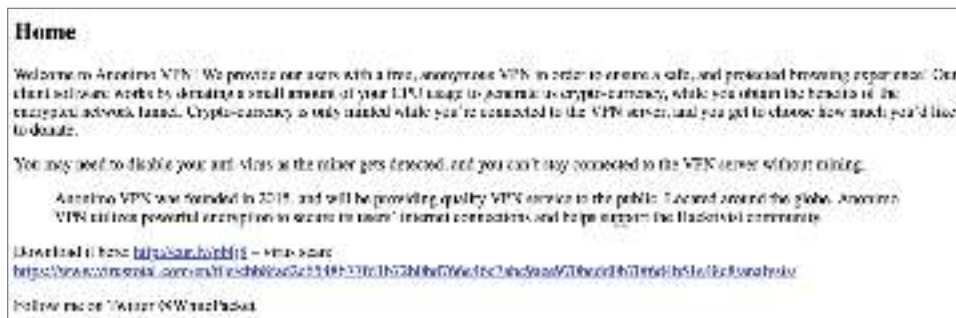


Figure 11.24

Now that we know the owner is the same, let's see if there is anything interesting in the website copy. Reading through the site, the website copy indicates that Anonimo.ninja is a VPN client that is also a Bitcoin miner. The download

link for “Anonimo VPN.exe” does not work, but luckily the Virustotal link is still active.

Figure 11.25 shows the result of searching for the Anonimo VPN.exe client on Virustotal.



Figure 11.25

The virustotal comments on this file (Figure 11.26) look suspicious.



Figure 11.26

Judging by the comments, it looks like our target’s VPN software was actually a Bitcoin miner (which we already knew from the site’s description). The good news is that the signature left us an IP address outside of Cloudflare: 199.168.137.147. A search on that IP address in Virustotal gives us the owner information shown in Figure 11.27.

199.168.137.147	
As Owner	VolumeDrive
ASN	46554
Country	US
Click to select	

Figure 11.27

Looking at the data, we can see that the ASN (and the host) is VolumeDrive (www.volumedriver.com). Nothing actionable from this information (yet), but it is important to note that the information may come up later.

A Major Find

The previous scenario was just one single thread of a possible investigation path. There was actually a fairly significant piece of information discovered within those results that I did not realize until much later.

When we were looking at the screenshot history for WhitePacket.com, there was a screenshot from 2016 of his old Twitter account. Figure 11.28 shows a zoomed-in version of that screenshot.



Figure 11.28

This find is *extremely subtle*, and it wasn't until I was writing the final chapters of this book that I realized it.

To put this into the proper context, the following are excerpts from a single conversation with TDO:

```
TDO:  We're not 'bros' at all, mate.
VT:   see, and in your email you said we were friends.
TDO:  You're fortunate, tonight.
VT:   But fine, i understand. i will keep this totally
      professional from now on. Didn't mean to offend
TDO:  I'm sitting here moving terabytes of data into a new
      server, and I'm bored, so you've been blessed with
      communicating with me.
TDO:  Yes, take a few moments to respect, you cuck.
```

```

TDO:    F*** off, where is your alcohol?
VT:     i will be out tomorrow night and will make sure i pick up
        some beer on my way home for our next chat
TDO:    You're such a f***** cuck.
TDO:    I'm the only one influenced by drugs here,
---
TDO:    You're site is SHITE.
TDO:    It gets hacked DAILY.
TDO:    You dumb cuck.
VT:     yeah by the way
VT:     who is Argon
TDO:    No, we're not answering.
VT:     because you dont know?
TDO:    Because we're not answering.

```

Did you notice anything?

Now back to WhitePacket, the description on his personal Twitter page reads. . .

“Lover of Bitcoin: weaboo/**cuck**/subhuman filth.” The important word here is “cuck.”

```
2:07 AM    TDO    You dumb cuck.
```

I have personally spent thousands of hours conversing with different threat actors both in private chats and on forums. I have not once—not one single time—ever heard (or seen) someone use the word “cuck.”

I am not an attorney, so I can’t speak to how significant a find this would be in a courtroom. However, I do know that a common technique used by law enforcement agencies involves looking for commonalities in linguistic behavior, such as the use of unique words and reoccurring grammatical errors (e.g., *their* vs. *there*).

Combined with all other evidence, I suspect these occurrences of the word “cuck” are actually fairly significant.

Summary

This chapter showcased the power and breadth of coverage available in Iris by DomainTools. I chose to showcase Iris, despite it being a premium tool, because of the immense value it provided me during my investigation. I can say, without any doubts, that I would not have been able to uncover so many clues and piece together so much historical domain ownership information without access to this tool.

This chapter highlighted a number of Iris's key features and walked through the process of using those features in order to uncover the owner of a domain name. The information provided should be enough for you to decide whether access to this type of premium tool is necessary for your organization.

The next chapter will mark the beginning of Part III of this book, where we will start "Digging for Gold" by investigating document metadata.

Digging for Gold

In This Part

Chapter 12: Document Metadata

Chapter 13: Interesting Places to Look

Chapter 14: Publicly Accessible Data Storage

This section will focus on looking for data in different and unusual places. As the title suggests, we will be digging for gold in this section, which means the focus will be on tools and techniques that can help uncover useful (and often hidden) information from often-overlooked corners of the Internet.

Document Metadata

Document metadata refers to information that is stored within a file and used to provide context or descriptions about that file. Metadata is often invisible and provides supporting information about the file in which it is stored.

Document metadata can include pieces of information such as the document's title, the software used to create it, the name of the author or organization where it was created, the name of the computer on which it was created, and the date and time the file was first created or modified.

In addition to basic metadata information, different file types can contain different types of metadata, which can also vary between the software used to create that file. The amount of metadata that is saved with a document ultimately depends on the software that was used to create the document.

Where things get really interesting is when you come across sensitive documents that have not been stripped of their metadata. The metadata within these documents (especially in photos) can contain incredibly sensitive information.

To give you an example, in 2016, there was a scandal involving leaked nude celebrity photos. The photos were stolen from the victims' personal accounts and leaked online. Upon analysis, the metadata within the photos contained very specific and identifiable information such as the camera (or phone) type, lens settings, date and time the photos were taken, and even its geolocation!

This type of information can provide a substantial piece of evidence under the right circumstances. A quick search online will also reveal several past

news stories where fugitives posted pictures to their social media accounts, and metadata from those documents ultimately aided in their arrest.

These are only a few examples. The important thing to remember is that all documents have some sort of metadata information saved within them. This chapter will explore the different tools we can use to find and extract metadata from different document types.

Exiftool

Exiftool is an independent command-line tool used for reading, writing, and editing meta-information in files. Supported metadata formats include EXIF, GPS, ID3, XMP, GeoTIFF, and the majority of digital cameras (including Canon, Fuji, Kodak, Nikon, and many more). In addition to all of the different meta-information formats supported by Exiftool, it can also read that meta-information from virtually every known file type including, but not limited to, PDFs, documents, spreadsheets, images, audio files, and video files.

You can download Exiftool at <https://www.sno.phy.queensu.ca/~phil/exiftool>.

In order to showcase the tool's potential, I am using sample images from Ianaré Sévi's GitHub repository of sample images for testing metadata retrieval: <https://github.com/ianare/exif-samples>.

The tool itself is extremely straightforward and does not have many options to showcase. Running the tool consists of recalling the file and including the filename to analyze:

```
exiftool Canon_40D.jpg
```

```
ExifTool Version Number      : 11.33
File Name                    : DSCN0027.jpg
File Size                   : 154 kB
File Modification Date/Time  : 2019:04:04 04:57:47-05:00
File Access Date/Time       : 2019:04:04 04:57:47-05:00
File Inode Change Date/Time  : 2019:04:04 04:57:47-05:00
File Permissions             : rw-r--r--
File Type                   : JPEG
File Type Extension         : jpg
MIME Type                   : image/jpeg
Exif Byte Order              : Little-endian (Intel, II)
Image Description            :
Make                        : NIKON
Camera Model Name           : COOLPIX P6000
Orientation                 : Horizontal (normal)
X Resolution                 : 300
Y Resolution                 : 300
Resolution Unit              : inches
```



```

Software                               : Nikon Transfer 1.1 W
Modify Date                           : 2008:11:01 21:15:09
Exif Version                           : 0220
Date/Time Original                     : 2008:10:22 16:44:01
Create Date                           : 2008:10:22 16:44:01
Subject Distance Range                 : Unknown
[...truncated...]
GPS Latitude Ref                       : North
GPS Longitude Ref                      : East
GPS Altitude Ref                       : Above Sea Level
GPS Time Stamp                         : 14:42:29.03
GPS Satellites                         : 05
GPS Img Direction Ref                  : Unknown ()
GPS Map Datum                          : WGS-84
GPS Date Stamp                         : 2008:10:23
Compression                           : JPEG (old-style)
Thumbnail Offset                       : 4560
Thumbnail Length                       : 5803
Image Width                           : 640
Image Height                           : 480
Encoding Process                       : Baseline DCT, Huffman coding
Bits Per Sample                       : 8
Color Components                       : 3
Y Cb Cr Sub Sampling                  : YCbCr4:2:2 (2 1)
XMP Toolkit                           : Public XMP Toolkit Core 3.5
Rating Percent                         : 0
Aperture                              : 4.1
GPS Date/Time                           : 2008:10:23 14:42:29.03Z
GPS Latitude                           : 43 deg 28' 6.39" N
GPS Longitude                         : 11 deg 52' 53.45" E
GPS Position                           : 43 deg 28' 6.39" N, 11 deg 52' 53.45" E
Image Size                             : 640x480
Megapixels                             : 0.307
Scale Factor To 35 mm Equivalent: 4.7
Shutter Speed                          : 1/148
Thumbnail Image                        : (Binary data 5803 bytes, use -b option
to extract)
Circle Of Confusion                    : 0.006 mm
Field Of View                          : 65.5 deg
Focal Length                           : 6.0 mm (35 mm equivalent: 28.0 mm)
Hyperfocal Distance                    : 1.36 m
Light Value                            : 11.9

```

Finding a fully loaded document like this during an investigation may seem unlikely, but the reality is that most criminals either are not tech savvy or just think they are invincible. There have been numerous reported cases where police have made arrests based on information recorded in a document's metadata, so you should not count this out as a possibility when searching for information!

Metagoofil

Metagoofil is an information-gathering tool that allows metadata to be extracted from publicly available documents on web servers. While Exiftool is great at reading metadata from files we already have, Metagoofil can help take your investigation further by allowing you to *find* files about your target.

Metagoofil works by performing dork searches across Google to find documents containing potentially useful metadata. Its capabilities include being able to save the documents locally and extract metadata remotely from popular file types and formats including (but not limited to) Word documents, Excel sheets, PDF files, and many more.

Names, email addresses, shared resources, and server names are just some of the metadata that can be extracted from documents using the tool.

You can download Metagoofil from <https://github.com/laramies/metagoofil>.

The parameters of Metagoofil include:

```
-d: domain to search
-t: filetype to download (pdf,doc,xls,ppt,odp,ods,docx,xlsx,pptx)
-l: limit of results to search (default 200)
-h: work with documents in directory (use "yes" for local analysis)
-n: limit of files to download
-o: working directory (location to save downloaded files)
-f: output file
```

Running Metagoofil is straightforward. Enter the command followed by `-d` and the domain you want to search. For example:

```
python metagoofil.py -d apple.com
```

Before running our search, I like to set up my working directory of files (so Metagoofil saves a copy of everything it finds) and create an output file so I can see my results. Metagoofil also requires you to specify which file types to look for (or it won't run). We will also decrease the search results to a maximum of 100 and set the limit of files to download at 100. Let's run a search with those parameters against Busey Bank and see what we can find:

```
python metagoofil.py -d busey.com -t doc,pdf,xls,docx,xls,xlsx -l 100
-n 100 -o docs -f results.txt
```

Wait until Metagoofil completes the search, and then you can view the results by opening the HTML file. The downloaded data might include user names, software versions, emails, servers and paths, and files and the metadata of each

file, including names of authors and the location of each downloaded file on the machine used to process the search request:

```
*****
*      /\ \  _ _| | _ _ _ _ _ _ _ _ _ _ / _ ( ) | *
*      /      \ / _ \ _ _/ _ \ | / _ \ | / _ \ | | _ | | *
*      / /\ \  _ _| | | | | | | | | | | | | | | | | | | *
*      \ \      \ \ _ _ \ \ _ \ | \ _ \ | \ _ \ | _ | | | *
*                                     | _ _/
*
* Metagoofil Ver 2.2
* Christian Martorella
* Edge-Security.com
* cmartorella_at_edge-security.com
*****
['pdf']
```

```
[-] Starting online search...
```

```
[~] Searching for pdf files, with a limit of 10
Searching 100 results...
```

Results: 106 files found

Starting to download 100 of them:

```
[1/10] /webhp?hl=en
[2/10] https://www.microsoft.com/buxtoncollection/a/pdf/
ebookman_manual.pdf
[3/10] http://go.microsoft.com/fwlink/p/%3Flinkid%3D528467
[4/10] http://nds1.webapps.microsoft.com/phones/files/guides/
Nokia_302_UG_es.pdf
[5/10] http://nds1.webapps.microsoft.com/phones/files/guides/
Nokia_100_UG_en.pdf
[6/10] http://nds1.webapps.microsoft.com/phones/files/guides/
6310i_usersguide_hu.pdf
[7/10] http://nds1.webapps.microsoft.com/phones/files/guides/
Nokia_6020_UG_es.pdf
[8/10] http://nds1.webapps.microsoft.com/phones/files/guides/
Nokia_100_UG_nl.pdf
[9/10] http://nds1.webapps.microsoft.com/phones/files/guides/
Nokia_1616_1800_UG_fr.pdf
[10/10] http://nds1.webapps.microsoft.com/phones/files/guides/
Nokia_6555_UG_nl.pdf
processing
```

```
[+] List of users found:
-----
mode

[+] List of software found:
-----
Acrobat Distiller 8.1.0 (Windows)
AH Formatter V5.3 R1 (5,3,2011,0425) for Windows
Acrobat Distiller 4.05 for Windows
FrameMaker+SGML 5.5.6p145
Acrobat Distiller 5.0.5 (Windows)
FrameMaker 6.0

[+] List of paths and servers found:
-----

[+] List of e-mails found:
-----
Acrobat Distiller 5.0.5 (Windows)
FrameMaker 6.0
Acrobat Distiller 5.0.5 (Windows)
FrameMaker 6.0
```

Recon-NG Metadata Modules

Recon-NG (initially discussed in Chapter 5) is a manual discovery tool with built-in modules designed to help quickly find and extract useful metadata from files and documents.

Metacrawler

Recon-NG's Metacrawler module searches for files associated with a given domain, and then extracts contact-related metadata from those files. To use the Metacrawler module within Recon-NG, type:

```
use recon/domains-contacts/metacrawler
```

Figure 12.1 shows the Metacrawler module information, which you can see by typing **show info**.

The Metacrawler module also has an `EXTRACT` option that will extract the metadata from the discovered files. If this is set to `False`, the module will display any contact information found within the searched files and display the results on screen—but will not extract (and save) the results.


```

[*] https://www.apple.com/environment/pdf/Apple_Facilities_Report_2020.pdf
[*] Title: untitled
[*] Creationdate: D:20090918122416-07'00'
[*] Producer: Acrobat Distiller 7.0 for Macintosh
[*] Moddate: D:20090918122416-07'00'
[*] https://www.apple.com/legal/dacs/US_IPSA.pdf
[*] Producer: IndirectObject(72, 0)
[*] Creator: IndirectObject(75, 0)
[*] Author: IndirectObject(73, 0)
[*] Title: IndirectObject(71, 0)
[*] Appl:Keywords: IndirectObject(78, 0)
[*] Moddate: IndirectObject(76, 0)
[*] Keywords: IndirectObject(77, 0)
[*] Creationdate: IndirectObject(76, 0)
[*] Subject: IndirectObject(74, 0)
[*] https://www.apple.com/accessibility/pdf/iPad_nano_7th_gen_VPAT.pdf
[*] Producer: IndirectObject(48, 0)
[*] Creator: IndirectObject(51, 0)
[*] Author: IndirectObject(49, 0)
[*] Title: IndirectObject(47, 0)
[*] Appl:Keywords: IndirectObject(54, 0)
[*] Moddate: IndirectObject(52, 0)
[*] Keywords: IndirectObject(53, 0)
[*] Creationdate: IndirectObject(52, 0)
[*] Subject: IndirectObject(50, 0)
[*] https://www.apple.com/environment/pdf/Apple_FY2016_Assurance_Statement.pdf
[*] Producer: Microsoft Word 2010
[*] Author: BY User
[*] Creator: Microsoft Word 2010
[*] Moddate: D:20170410080929-07'00'
[*] Title: ASR TT-06
[*] Creationdate: D:20170410080929-07'00'
[*] Subject: Assurance of Sustainability Reports - Template Assurance Statement (Medium)
[*] https://www.apple.com/newsroom/pdf/q2017data_xxx.pdf
[*] Producer: Mac OS X 10.5.6 Quartz PDFContext
[*] Creator: Microsoft Excel
[*] Author: Faraz Fatahi
[*] Title: 09-04-22 *Data Summary Excel.xls
[*] Appl:Keywords: [u'']
[*] Moddate: D:2009042216370000'00'
[*] Creationdate: D:2009042216370000'00'

```

Figure 12.3

After running the module, the extracted data will be located in the contacts table. The following command will show the extracted data:

```
show contacts
```

If Recon-NG displays the “no data returned” message, that means that the module was not able to gather any contact metadata from the files. For legal reasons, we cannot display other people’s scraped email addresses, and displaying a screenshot with a bunch of black boxes through it does not seem useful. So let’s move on.

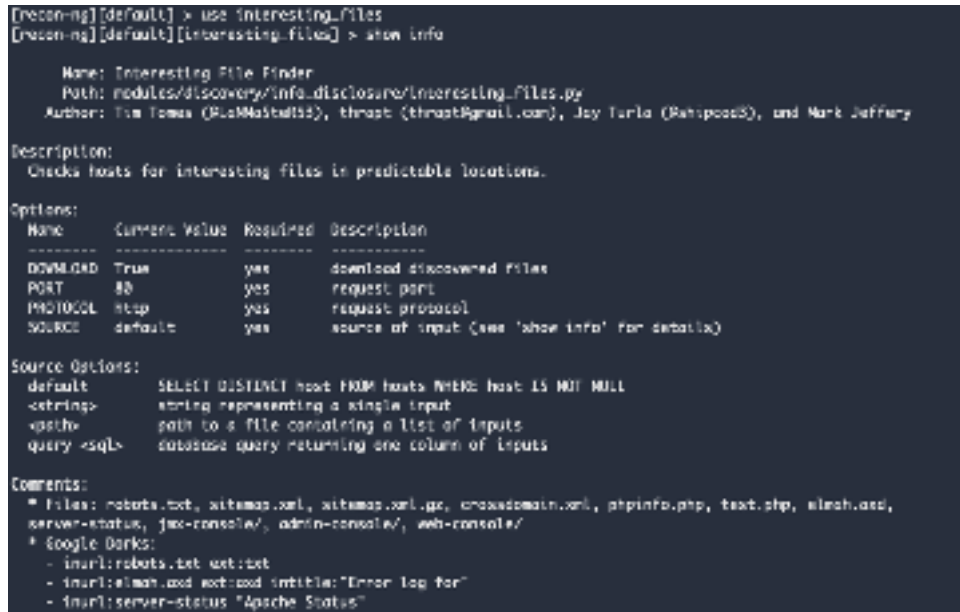
Interesting_Files Module

“Interesting_files” is one of Recon-NG’s built-in discovery modules. As the name suggests, the module is used for “finding interesting files in predictable locations.”

If you're not sure how to find a particular module in Recon-NG, you can search for it by typing `search` followed by the module name. The following command will search for the `interesting_files` module:

```
search interesting_files
```

To load the module, type `use interesting_files`. Figure 12.4 shows the output of the `show info` command when run after loading the `interesting_files` module.



```
[recon-ng][default] > use interesting_files
[recon-ng][default][interesting_files] > show info

Name: Interesting File Finder
Path: modules/discovery/info/disclosure/interesting_files.py
Author: Tim Tomes (RiskMaster133), thrust (thrust@gmail.com), Jay Taria (RahipooS), and Mark Jeffery

Description:
  Checks hosts for interesting files in predictable locations.

Options:
  Name      Current Value  Required  Description
  -----
  DOWNLOAD  True             yes        download discovered files
  PORT      80               yes        request port
  PROTOCOL  http             yes        request protocol
  SOURCE     default           yes        source of input (see 'show info' for details)

Source Options:
  default      SELECT DISTINCT host FROM hosts WHERE host IS NOT NULL
  <string>      string representing a single input
  <path>        path to a file containing a list of inputs
  query <sql>   database query returning one column of inputs

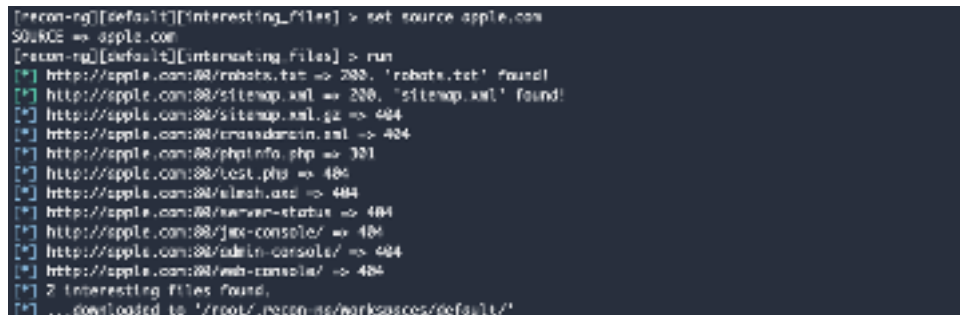
Comments:
  * Files: robots.txt, sitemap.xml, sitemap.xml.gz, crossdomain.xml, phpinfo.php, test.php, slash.asp,
  server-status, jmx-console/, admin-console/, web-console/
  * Google Dorks:
    - inurl:robots.txt ext:txt
    - inurl:slash.asp ext:asp intitle:"Error log for"
    - inurl:server-status "Apache Status"
```

Figure 12.4

As we can see from the screenshot, the module needs info from the `hosts` table before we can run it. For our example, we will set the source to `Apple.com`:

```
set source apple.com
```

Figure 12.5 shows the output after running the module on `Apple.com`.



```
[recon-ng][default][interesting_files] > set source apple.com
SOURCE => apple.com
[recon-ng][default][interesting_files] > run
[*] http://apple.com:80/robots.txt => 200, 'robots.txt' found!
[*] http://apple.com:80/sitemap.xml => 200, 'sitemap.xml' found!
[*] http://apple.com:80/sitemap.xml.gz => 404
[*] http://apple.com:80/crossdomain.xml => 404
[*] http://apple.com:80/phpinfo.php => 301
[*] http://apple.com:80/test.php => 404
[*] http://apple.com:80/slash.asp => 404
[*] http://apple.com:80/server-status => 404
[*] http://apple.com:80/jmx-console/ => 404
[*] http://apple.com:80/admin-console/ => 404
[*] http://apple.com:80/web-console/ => 404
[*] 2 interesting files found,
[*] ...downloaded to '/root/.recon-ng/workspaces/default/'
```

Figure 12.5

There were two interesting files discovered on this domain. The `robots.txt` file is especially interesting because this is the file that tells search engines which folders to index and which to ignore.

In other words, if a website has folders that people (or search engines) should not be looking at, those folders would be notated in this file. The discovery of folders that should not be indexed is typically a good sign that you are on the right track to discovering something interesting; those folders are almost always worth exploring.

Pushpin Geolocation Modules

The Pushpin modules in Recon-NG look for files that have been geotagged, or that have geolocation metadata saved within them. The name is derived from the pushpins that are pinned to maps showing locations. In this case, these modules are designed to do exactly that—to pinpoint the exact location from which the files were posted.

With the level of security included in most social media and online sites, stumbling upon an online image or file with embedded location data is akin to hitting the lotto—it is *extremely* rare and will probably only happen under the right circumstances. Nevertheless, plenty of scenarios exist where this could be a common find, such as during the direct examination of a phone's archive, so you should always try.

The Pushpin module searches Flickr, Shodan, Twitter, and YouTube for media with embedded geolocation data. The Pushpin module will *not* tell you where a specific piece of media was published. Instead, you can enter an address or specific coordinates and the module will return all media published around that location.

Annoyingly, each of those sites has its own module, so you need to search them individually.

To search for the location of the Pushpin modules, type the following:

```
search pushpin
```

```
[recon-ng] [default] > search pushpin
[*] Searching for 'pushpin'...
```

```
Recon
-----
recon/locations-pushpins/flickr
recon/locations-pushpins/shodan
recon/locations-pushpins/twitter
recon/locations-pushpins/youtube
```

```
Reporting
-----
reporting/pushpin
```


Each module has its own set of parameters and descriptions, which you can view by loading the module and using the `show info` command. Let's get info on the Flickr module:

```
[recon-ng][default] > use flickr
[recon-ng][default][flickr] > show info
```

```
Name: Flickr Geolocation Search
Path: modules/recon/locations-pushpins/flickr.py
Author: Tim Tomes (@LaNMaSteR53)
Keys: flickr_api
```

Description:

Searches Flickr for media in the specified proximity to a location.

Options:

Name	Current Value	Required	Description
RADIUS	1	yes	radius in kilometers
SOURCE	default	yes	source of input (see 'show info' for details)

Source Options:

```
default      SELECT DISTINCT latitude || ', ' || longitude FROM
locations WHERE latitude IS NOT NULL AND longitude IS NOT NULL
<string>     string representing a single input
<path>       path to a file containing a list of inputs
query <sql>  database query returning one column of inputs
```

Comments:

* Radius must be greater than zero and less than 32 kilometers.

We can see from the module's information page that we need to have the coordinates of an address as well as the address itself before we can use the module.

Plenty of free tools are available online to do this. One example is https://www.mapdevelopers.com/geocode_tool.php. After visiting the website, enter your desired address, and it will return your latitude and longitude coordinates.

For this example, I am using the coordinates of St. Louis, MO. The MapDeveloper.com website returned the following info:

```
Latitude 38.6337716
Longitude -90.2416548
```

Now that we have location coordinates, let's see what kind of information we can find in the Pushpin module.

First, we need to add the location to Recon-NG using the `add locations` command:

```
[recon-ng][default][geocode] > add locations
latitude (TEXT): 38.6337716
```

```
longitude (TEXT): -90.2416548
street_address (TEXT):
```

This is where things can get really fun. The latitude and longitude of St. Louis have been entered as location data—now let's see if anyone from this area has tweeted recently. Executing this search involves loading the Twitter Pushpin module using the following command:

```
use recon/locations-pushpins/twitter
```

Figure 12.6 shows a sample of the results found after running the module.

```
(recon-ng)[default] > use recon/locations-pushpins/twitter
(recon-ng)[default](Twitter) > run

-----
38.6337715, -90.2416548
-----
[*] Collecting data for an unknown number of tweets...
[*] Latitude: 38.6337715
[*] Longitude: -90.2416548
[*] Media URL: https://twitter.com/Bethany2125/statuses/1114549113862296833
[*] Message: Friday at the Fall 🍂💙🍷 #winerecushol @ The Fabulous Fox Boys // @/08/15/2019
[*] Profile Name: Bethany
[*] Profile URL: https://twitter.com/Bethany2125
[*] Screen Name: Bethany2125
[*] Source: Twitter
[*] Thumb URL: https://pbs.twimg.com/profile_images/1116284762514338944/Uq2tbc_normal.jpg
[*] Time: 2019-08-06 01:47:47
-----
[*] Latitude: 38.6337715
[*] Longitude: -90.2416548
[*] Media URL: https://twitter.com/AL_Elwoods/statuses/1114331713357116017
[*] Message: @pops: let's be Legendary @hostest servers especially the ones in the Lou. @danielc_hense @Cadea_Rubio
[*] Profile Name: Al Elwoods
[*] Profile URL: https://twitter.com/AL_Elwoods
[*] Screen Name: AL_Elwoods
[*] Source: Twitter
[*] Thumb URL: https://pbs.twimg.com/profile_images/86897146823276895/1h1h1d_normal.jpg
[*] Time: 2019-08-06 00:59:18
-----
[*] Latitude: 38.6337715
[*] Longitude: -90.2416548
[*] Media URL: https://twitter.com/AL_Elwoods/statuses/1114331713357116017
[*] Message: PARTY FIRST SATURDAY EVER222 #BIR222A-YEPNOM #OU #DULIFE #DULIFE222 #WEMOUSTIL #OTBMC SHOOTOUT
[*] Profile Name: Al Elwoods
[*] Profile URL: https://twitter.com/AL_Elwoods
[*] Screen Name: AL_Elwoods
[*] Source: Twitter
[*] Thumb URL: https://pbs.twimg.com/profile_images/86897146823276895/1h1h1d_normal.jpg
[*] Time: 2019-08-06 00:56:18
-----
```

Figure 12.6

This is pretty powerful stuff when you think about it. I set the location data to a random spot in St. Louis. But what if we were looking for activity coming from a particular target, like a company?

If the goal is to perform recon and intelligence gathering on a company to gather information that might allow us to either gain access or target a specific employee, a good place to start is social media.

Keeping on with our example, Anheuser-Busch happens to be one of the largest companies in St. Louis. Using the MapDevelopers website, a quick search for Anheuser-Busch St. Louis returns the following coordinates:

Latitude: 38.62727

Longitude: -90.19789

Figure 12.7 shows some of the 1,200+ tweets discovered after entering those coordinates into Recon-NG and re-running the Twitter Pushpin module.

```

-----
[*] Latitude: 38.6272
[*] Longitude: -90.1978
[*] Media_URL: https://twitter.com/steph_m29/statuses/1118918911848671767
[*] Message: From tiny baby to big boy pup in only 4 weeks! 🐶 @ St. Louis https://t.co/mr5eS4jH0n
[*] Profile_Name: Stephanie Paine
[*] Profile_URL: https://twitter.com/steph_m29
[*] Screen_Name: steph_m29
[*] Source: Twitter
[*] Thumb_URL: https://pbs.twimg.com/profile_images/929907869/842/6592/01-fdw5_normal.jpg
[*] Time: 2019-09-27 14:58:02
-----
[*] Latitude: 38.627025
[*] Longitude: -90.1994042
[*] Media_URL: https://twitter.com/bnj_sbl_sales/statuses/1118916201666443570
[*] Message: Can you recommend anyone for this job? Account Manager - https://t.co/7M3qrPzrX #StLouis, MO #Sales
[*] Profile_Name: Bnj-Sbl Sales Job
[*] Profile_URL: https://twitter.com/bnj_sbl_sales
[*] Screen_Name: bnj_sbl_sales
[*] Source: Twitter
[*] Thumb_URL: https://pbs.twimg.com/profile_images/660727751200126752/pDQyQn2_normal.jpg
[*] Time: 2019-09-27 14:46:34
-----
[*] Latitude: 38.627025
[*] Longitude: -90.1994042
[*] Media_URL: https://twitter.com/InvitationJobs/statuses/1118915424688049152
[*] Message: Can you recommend anyone for this job in St. Louis, MO? https://t.co/0PRn0Tfyk #Engineer #TechLife
[*] Profile_Name: Invitation, LLC
[*] Profile_URL: https://twitter.com/InvitationJobs
[*] Screen_Name: InvitationJobs
[*] Source: Twitter
[*] Thumb_URL: https://pbs.twimg.com/profile_images/1254948620/InvitationBuilding-normal.jpg
[*] Time: 2019-09-27 14:44:14
-----
[*] Latitude: 38.6286282
[*] Longitude: -90.2020457
[*] Media_URL: https://twitter.com/TinaSTL/statuses/1118903288371866736
[*] Message: Ward 8 Downtown St. Louis https://t.co/5o63jg_mWg
[*] Profile_Name: TinaSTL
[*] Profile_URL: https://twitter.com/TinaSTL
[*] Screen_Name: TinaSTL
[*] Source: Twitter
[*] Thumb_URL: https://pbs.twimg.com/profile_images/509578518007404012/nc37Q2ZF_normal.jpg
[*] Time: 2019-09-27 14:20:39
-----
[*] 1428 tweets @P0005566.

SUMMARY
-----
[*] 1200 total (1200 new) pushpin found.

```

Figure 12.7

Intrigue.io

If you are interested in a one-stop-shop OSINT scanning/discovery/metadata extraction tool, Intrigue.io may be what you are looking for. Previously discussed in Chapter 5, Intrigue.io is a behemoth all-in-one tool that tries to completely automate the process of OSINT discovery. Intrigue.io's metadata discovery and extraction process is no different.

In order to discover and extract metadata from our target, our first step will be to use the URI Spider module. From the menu, click the Start button, then change the Task option to URI Spider.

In the new task screen, set URI Spider with the initial host target set to our target. Included in the options are the spider depth (default is 10), and a limit to the maximum number of pages to spider. The default configuration will also extract DNS Record types, PhoneNumbers, and EmailAddress type entities in the content of the page. When looking to extract metadata, it is important to set the extract-uris option to true. Figure 12.8 shows the URI Spider screen with these options set this way.

Start

Task	URI Spider
Entity Type	uri
Entity Name	http://www. [redacted]
limit	100
max_depth	10
spider_whitelist	{current domain}
extract_dns_records	true
extract_dns_record_pattern	{current domain}
extract_email_addresses	true
extract_phone_numbers	true
parse_file_metadata	true
extract_uris	true
user_agent	Mozilla/5.0 (Macintosh; Intel Mac OS X 10_10_2) AppleWebKit
Machine	Org Asset Discovery (Active) 'Machine' specifies the post-processor for each new entity.
Iterations	3 iterations 'Iterations' specifies the depth to which the machine is run.
	☒ Auto Enrich

Run Task

Figure 12.8

The spider will identify hundreds of file types and parse their content and metadata. Intrigue.io supports over 300 file formats, including common formats like DOC, DOCX, and PDF—as well as more exotic types like application/ogg and many video formats.

When the task is executed, the screen will change to the task result page (Figure 12.9). This page will show you a running list of the discovered entities. The terminal window to the left will also show the progress of the scanning and extraction process.



Figure 12.9

When complete, the terminal window will say “task complete.” To view the results of our scan, click the Entities button in the main navigation bar. On the entities page, the statistics area (Figure 12.10) will summarize all of the different entities that have been discovered and extracted from the documents’ metadata.

Looking at the results summary, this is actually a very impressive find. The results include 31 domains, 1 phone number, 5 names, and 2 AWS S3 buckets!

Intrigue.io will also automatically attempt to access the discovered buckets. To view information on the event, click the AWS Bucket link in the statistics area. Figure 12.11 shows the entity list screen.

From here, click the bucket you want to explore. Figure 12.12 shows the individual entity page with the list of items discovered (and downloaded) from the AWS bucket.



Figure 12.10

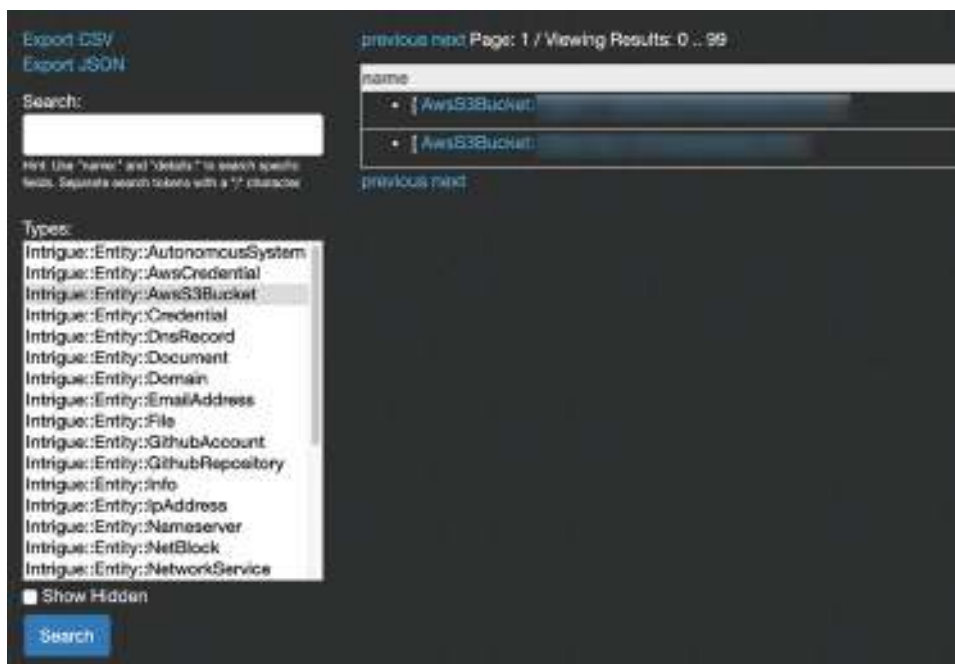


Figure 12.11

Discovered bucket information can be a significant find. An open AWS bucket for your target can be a great place to poke around and look for leaking data. AWS buckets are not easy to properly secure, which is why plenty of organizations have accidentally left their buckets unsecured or set with inadequate permissions. They are always a great place to look.

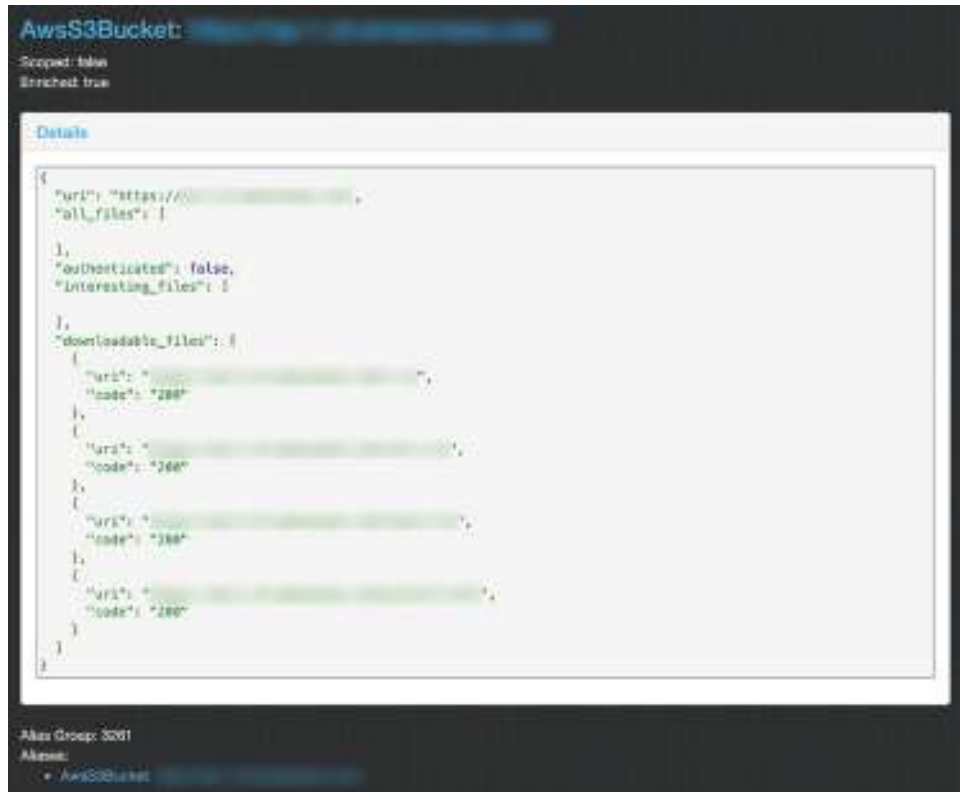


Figure 12.12

FOCA

Fingerprinting Organizations with Collected Archives, or FOCA, is an open-source reconnaissance-gathering tool for Windows. FOCA is a GUI tool that downloads publicly available documents from web servers and scans them for metadata and other hidden information. FOCA uses Google, Bing, and DuckDuckGo to search for documents.

FOCA can analyze various document formats, including (but not limited to) PDF, DOC, XLS, PPT, JPG, PNG, SVG, and more. It can also be used to extract EXIF information from graphic files. A useful feature of FOCA is that it analyzes the document metadata prior to downloading the file and limits the downloads to only files that contain useful metadata information.

In addition, FOCA has a server discovery module that allows you to search for vulnerabilities in documents and hosts. The app can also automatically search for other hosts and domain names related to the main target domain by applying its own DNS-based reconnaissance techniques, such as reverse DNS

lookups and PTR log scans to find other related servers in the same address segment. In addition, FOCA can perform dictionary bruteforcing of DNS to find hidden subdomains.

FOCA also has its own market where users can extend the app's functionality using third-party plugins. You can download FOCA from <https://github.com/ElevenPaths/FOCA>.

Starting a Project

When FOCA opens, the first step requires setting up a project. Figure 12.13 shows the initial FOCA screen with Microsoft listed as our project name and Microsoft.com listed as our target domain.



Figure 12.13

One of the really nice features of FOCA is that it will automatically organize your projects by domain (which can be found in the domains section). Researchers who are constantly analyzing metadata should find this especially useful because of how organized it can make the document collection process.

After FOCA creates the project, our next step in metadata discovery is to click the Metadata tree item. Figure 12.14 shows the empty metadata area after clicking the tree item. FOCA automatically selects all file extensions for our metadata search. On the same screen, you can also expand the search area outside of just Google by selecting Bing and DuckDuckGo as additional search engines in the “Search engines” section near the top right, next to the Extensions section.

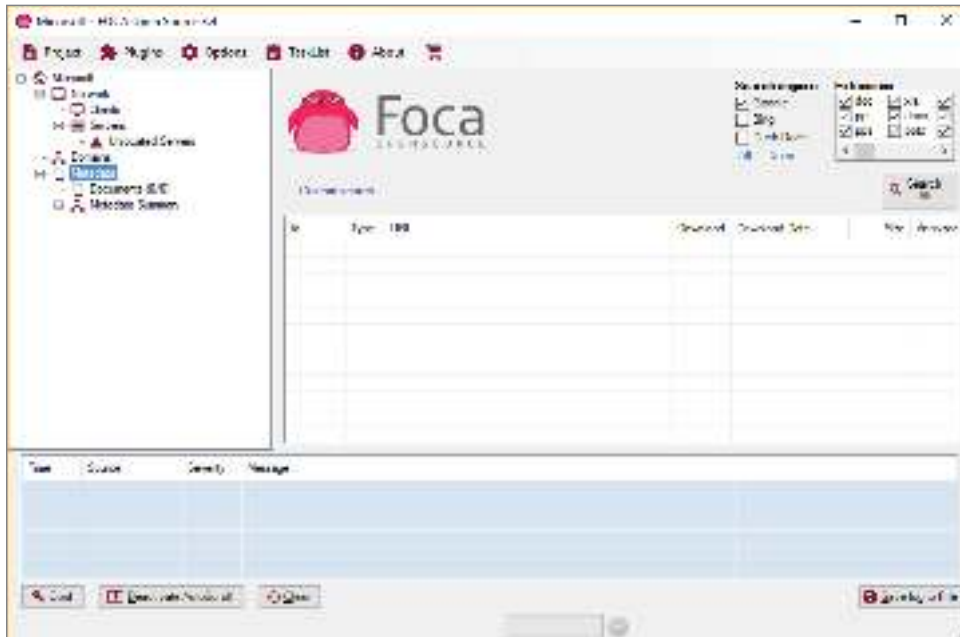


Figure 12.14

NOTE Don't forget to specify your API keys in the settings. This type of searching will require a lot of hits to Google and other search engines, which will almost certainly be flagged or eventually blocked by Captcha. Running Google or Bing searches using an API key will ease some of those restrictions.

When you are ready to kick off the scan, click the Search All button. Any discovered documents will be displayed within the grid area.

All discovered documents are initially marked with an x in the download column. This is to indicate that the file has not been downloaded yet. Right-clicking any of the documents will display a menu of actions that can be performed with the documents (shown in Figure 12.15).

FOCA requires you to first download a document before being able to analyze or extract any metadata. If a document has not been downloaded, it will display an x in the download column. After downloading a document, the x becomes a ●, signifying the file has been successfully downloaded (shown in Figure 12.16).

Extracting Metadata

After the documents have been downloaded, you have the ability to extract any metadata stored within them. To analyze the metadata stored within the files, first click Extract Metadata, followed by Analyze Metadata (refer to Figure 12.15). If metadata is found in a file or files, the Metadata folder is updated accordingly.

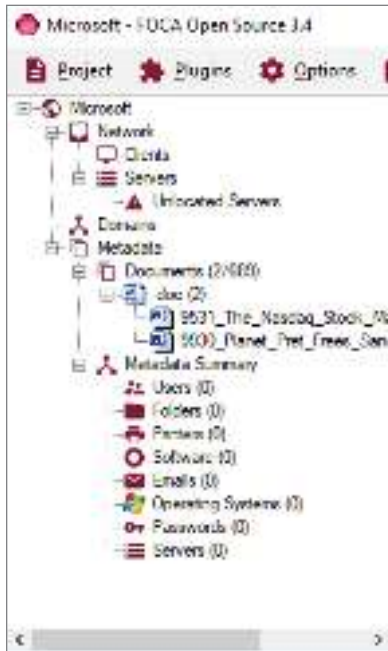


Figure 12.17

Extracted metadata will be stored within nicely organized sub-tree lists. For example, Figure 12.17 shows the documents tree with a DOC sub-tree. Additional sub-trees will appear as new document types are detected (such as PDF, XLS, etc.). Possible extracted metadata includes user names, folder locations and server names, printer information, software titles, email addresses, and passwords.

FOCA allows users to manually add a file (or folder containing several files) to the list on the grid to auto-extract their metadata. You can also add a list of URLs from a file in order to instruct FOCA to gather information on those URLs.

After the metadata extraction process is complete, each file will have numbers displayed next to them. These numbers indicate how many pieces of metadata were discovered in each document.

Any of this information can easily be used to carry out targeted attacks against an organization. For example, any detected names can be used to deduce a user's login name. From there, a credential lookup site (discussed more in Chapter 18) can be used with the combined results to carry out a password spray attack against an organization's web login panels (e.g., Office 365).

Following a quick search of the first 100 files on Microsoft.com, Figure 12.18 shows the different pieces of extractable metadata.

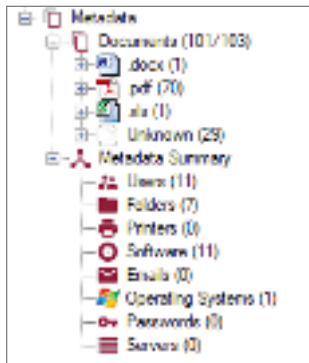


Figure 12.18

Summary

This chapter looked at document metadata, which is information located within a file that is used to describe the contents of that file. This chapter also discussed different file types, the type of sensitive information that can potentially be left behind within a document, and tools that can be used to automatically discover and extract potentially sensitive metadata information from publicly accessible documents.

The tools in this chapter vary in terms of complexity and base feature sets and therefore have their own sets of pros and cons. For example, I would consider tools like Exiftool and Metagoofil to be your “quick and dirty” solution if you need to get something done quickly.

On the other hand, FOCA is a tool designed purely for document metadata analysis. If your goal is to only analyze metadata (or make sure you are performing the most thorough analysis possible), this would be the tool to use. Intrigue.io provides a good middle ground for analysis capabilities while also bundling in a huge array of other features.

The next chapter will focus on unique interesting places to look for potentially sensitive (or earth-shattering) information.

Interesting Places to Look

As the name implies, this chapter is all about finding information in unique places. This chapter will also wrap up the story of threat actor Cyper and the events leading up to the discovery of his identity.

What better way to start than with an excerpt of a private XMPP conversation between Cyper and myself. In this conversation, Cyper and I traded information as to how we were able to discover each other using obscure clues.

```
VT:      How did you know it was me?
kickass: lol you use the same macbook
VT:      considering i bought this macbook 2 days ago, i dont
          think so
VT:      brand new baby. i9
kickass: but not the name ;)
kickass: anyway answer the questions
kickass: why you think i am that guy
VT:      ignoring the fact that your language style changes
          somewhere towards the end of hell/right before
          BlackBox, when you guys finally opened KA, your newsbot
          was something like Cypernews
VT:      CYPERCRIME news
VT:      that's what it was
```

```
kickass: lol
kickass: you are on ka and so you should know cyper is also
        there - btw have we ban you?
VT:      i havent been on KA in a long time
kickass: some tip if you use jabber don't use the same userpic
        for all accounts and also change you laptop name
        sometimes ;)
```

What Cyper was referring to was the “hostname” field in my Jabber client. I didn’t realize this at the time, but you have the ability to customize that field, and if left alone, Adium for Mac OSX will use the name of your computer. In this case Cyper knew he was speaking with me because the name of my computer was unique enough to identify.

Well done, and that’s exactly the type of obscure information we will be looking for in throughout this chapter.

TheHarvester

TheHarvester is an open-source OSINT tool that gathers publicly available email addresses, subdomains, IPs, and URLs from a wide range of data sources, including Baidu, Bing, Censys.io, Crt.sh, Dogpile, Google, LinkedIn, NetCraft, PGP, ThreatCrowd, Twitter, and VirusTotal.

This is probably one of the best command-line reconnaissance tools available because it covers such a wide search area. I would go so far as to say that if it doesn’t turn up any results, then you aren’t using it right.

TheHarvester uses a number of techniques to find information about its targets, including DNS bruteforce attacks using dictionary enumeration, DNS reverse lookups, hostnames, and of course, search engine dorking.

NOTE I went back and forth about which chapter to include this tool in because it is so versatile. This chapter felt like the best fit because theHarvester covers much more than the topics we discussed in the previous chapters, and I did not want to leave out any features that did not fit within the scope of those chapters.

I personally use theHarvester when I am looking to broadly scrape publicly available information on company names and email addresses that are lingering on search engines.

You can download theHarvester from <https://github.com/laramies/theHarvester>.

The following is a list of arguments available in theHarvester:

```
-h, --help          show this help message and exit
-d DOMAIN, --domain DOMAIN
```

```

                                company name or domain to search
-l LIMIT, --limit LIMIT
                                limit the number of search results, default=500
-S START, --start START
                                start with result number X, default=0
-g, --google-dork               use Google Dorks for Google search
-p PORT_SCAN, --port-scan PORT_SCAN
                                scan the detected hosts and check for Takeovers
                                (21,22,80,443,8080) default=False, params=True
-s, --shodan                    use Shodan to query discovered hosts
-v VIRTUAL_HOST, --virtual-host VIRTUAL_HOST
                                verify host name via DNS resolution and search
                                for virtual hosts params=basic, default=False
-e DNS_SERVER, --dns-server DNS_SERVER
                                DNS server to use for lookup
-t DNS_TLD, --dns-tld DNS_TLD
                                perform a DNS TLD expansion discovery,
                                default False
-n DNS_LOOKUP, --dns-lookup DNS_LOOKUP
                                enable DNS server lookup, default=False,
                                params=True
-c, --dns-brute                 perform a DNS brute force on the domain
-f FILENAME, --filename FILENAME
                                save the results to an HTML and/or XML file
-b SOURCE, --source SOURCE
                                baidu, bing, bingapi, censys, crtsh, cymon,
                                dnsdumpster, dogpile, duckduckgo, google,
                                google- certificates, hunter, intelx, linkedin,
                                netcraft, securityTrails, threatcrowd, trello,
                                twitter, vhost, virustotal, yahoo, all
-x EXCLUDE, --exclude EXCLUDE
                                exclude options when using all sources

```

Running a Scan

Search engines typically frown on people scraping data, so theHarvester will insert time delays between requests to avoid detection. You can also limit the number of requests by limiting the number of return results. This will also allow you to work with a more manageable list. When saving your output, theHarvester will always output your results in HTML and XML formats.

Running a scan with theHarvester can be a little quirky.

WARNING If you do not input the required parameters when running a scan, theHarvester will appear to run a scan with 0 results.

In order for theHarvester to kick off a scan, you *must* use the domain (-d), limit (-l), and source (-s) parameters. If you do not include all three parameters, you will see the following results:

```
[ root@osint ] > ./theHarvester.py -d apple.com
```

[illegible]

```
[*] No IPs found.
```

```
[*] No emails found.
```

```
[*] No hosts found.
```

If you receive a message that says no IPs, emails, or hosts found, you most likely did not enter the parameters correctly.

The `-d` parameter is how you specify the domain or company name you want to search. The `-l` parameter is the limit on how many searches you want to perform against each search engine. The `-b` parameter is how you specify the source search engine for your search. This can include sites like Censys, crt.sh, Google, Bing, or Yahoo, to name a few. If you don't want to limit your search to any specific search engine, you *must* enter `-b all`.

WARNING Keeping your search limit high will get you temporarily banned from Google or any of the other search engines. The default search value is 500, which means theHarvester is searching through the first 500 search results. Google shows 10 search results per page, which means theHarvester will quickly be looking through 50 pages. That is certainly enough for Google (or any search engine) to detect you as a scraper and either throw up a Captcha notice or simply block your results. To avoid this, I recommend using a proxy service that will send every request through a different IP (e.g., Storm Proxies)

Now that we understand the importance of manually specifying the -d, -b, and -l parameters, let's kick off a scan against our friends at WhitePacket Security (www.whitepacket.com):

```
root@osint > ./theHarvester.py -d whitepacket.com -l 100 -b all
```

[illegible]

[*] Target: whitepacket.com

```
[*] Searching Bing.
```

```
[*] Searching Censys.
```

Searching IP results page 4.

[*] Searching Yahoo.

Searching 100 results.

[*] Searching Baidu.

Searching 100 results.

[*] Searching Google.

Searching 100 results.

[truncated]

Users from Twitter: 3

.....

[removed]

```
[*] IPs found: 1
```

[removed]

```
[*] Emails found: 2
```

[removed]

```
[*] Hosts found: 5
-----
www.whitepacket.com:104.24.118.111
[removed]
```

The results are a good start, especially for performing initial recon on a lesser known domain. TheHarvester found five hosts, two emails, and three Twitter users. Not bad, but now let's try one more test on a much larger target—Microsoft.com:

```
root@osint /opt/theHarvester > ./theHarvester.py -d microsoft.com -l
100
-b all
```

```
*****  
*                                     *  
*   _ | _ | _ _ _ _ / \ / \ _ - - _ _ _ _ _ | _ | _ _ _ - - | *  
*   | _ | _ _ \ / _ \ / / \ / / _ | ' - \ _ / / _ _ \ _ | _ / \ ' _ | *  
*   | _ | _ | | _ | _ / _ / ( | | \ v / _ \ _ \ || _ | _ | *  
*   \ _ | _ | _ \ _ | \ / / \ \ _ , _ | | \ / \ _ | | _ \ _ | _ | *  
*                                     *  
* theHarvester 3.0.6 v380 *  
* Coded by Christian Martorella *  
* Edge-Security Research *  
* cmartorella@edge-security.com *  
* *  
*****
```

[*] Target: microsoft.com

```
[*] Users found: 34
-----
[removed]
```

```
Users from Twitter: 24
-----
[removed]
```

```
[*] IPs found: 11
-----
[removed]
```

```
[*] Emails found: 18
-----
[removed]
```

```
[*] Hosts found: 2771
-----
[removed]
```

```
[*] URLs found: 131
-----
https://trello.com/..[removed]
```

As you can see, this is a much more impressive find, especially considering these results came from a single tool: 2,771 hosts, 18 emails, 34 users, 24 Twitter accounts, and 131 Trello URLs associated to Microsoft.

From here, it will be up to you on which way you go for further exploration. Just make sure you are documenting everything you find so you can easily reference it later.

Now let's see what else we can kick up around the Internet.

Paste Sites

Hackers and other threat actors will often use one or more paste sites to post a sample of their goods online. Depending on your goals, there is often a significant amount of information to be found on these sites. Paste sites will often contain samples of data breaches, messages from threat actors, and various doxes on individuals.

Popular paste sites include:

- Pastebin.com
- 0bin.net
- Doxbin.org
- Justpaste.it

Paste sites, like forums, are widely used among threat actors and can often provide pivotal information during an investigation. The problem with these sites is that specific pastes will often be taken down if they are found to violate the site's terms of service by containing PII (personally identifiable information) or other forms of private information.

As a result, plenty of organizations scrape these paste sites every day for new content. I do this as well, and will discuss how this can be done fairly easily later in this chapter.

Psbdmp.ws

As I mentioned earlier, pastes that violate terms of service by posting PII or other elicited information are quickly removed. This is unfortunate if the paste contained clues or evidence that we need for an investigation. Lucky for us, there is psbdmp.ws.

For a long time, psbdmp was one of my secret weapons because it is the only site that keeps a full historical archive of every paste going as far back as 2015. Not only that, its API is very user friendly and can be easily queried from the URL or an external application. Before I created an engine to scrape all paste sites for my DataViper platform, the psbdmp API was a key part of my investigation toolkit.

Forums

Forums are, in my opinion, a gateway to the underground crime market. What I mean by this is that being on many of these hacker forums is like a rite of passage for more advanced threat actors. There is a point in every young hacker's life when they were inevitably considered a skid (script-kiddie). Skids, like most children, act without thinking and in doing so will often leave a trail that somehow leads back to their real identity.

That is why, in my humble opinion, the true art of threat actor identification will come down to a researcher's ability to gather and search historical information.

Searching forums should not be limited to just hacker and dark web (TOR) forums. Even threat actors hang out on nonhacker forums, which can come in all shapes and sizes. Researching different forums can often provide you pivotal clues, such as linking aliases to new email addresses.

One such forum that has helped me in this regard has been the BitcoinTalk forum. I can't explain why, but I have found that a lot of threat actors I researched in the past seem to hang out there. My point here is that you should not limit your search category to one particular type of forum or site. Always expand your searches as wide as possible.

EXPERTTIP: CHRIS ROBERTS

When I ran my own labs we would build our own scrapers. We would start off with a set of URLs that were of interest. That could've been anything from paste sites to URLs of interest . . . I mean you know how much stuff is out there on that side of it.

It was very much a case of saying, "Okay, which ones are actually going to be useful?" It was everything from going back to the early stuff to also pulling in from breach data. We used a bunch of stuff with breach data.

You look at that, you look at the ability to pull the data off the darker side of the world. Again, it's the same kind of concept. It's one URL leads to 10 URLs or 20 URLs. You do a quick, high-level analysis of them, and you decide what prioritization you're gonna put on scraping them, based off of word searches, heat maps, and a bunch of other kinds of stuff.

Then, from there, the data gets scraped, it gets indexed inbound. You really end up building your own version of a very, very high DT-search type of environment. In this

case, we used Elasticsearch, and no DBMS architecture on the other end of it to make sure that the bloody thing would work properly.

And to acquire the data, more often than not it just takes reading what other people are saying. Let's just say you go to a forum, it doesn't matter what kind of a forum it is. If it's talking about hacking or cracking or torrents . . . You go to those forums and it'll lead you to another URL, or another website, and another one and another. So we start with X, Y, Zed forum, let's take a look and see if we have anybody with an account, user ID, or anything that would be useful to harvest. You can run that straight into a database. If your database is configured, it'll probably tell you exactly whose user accounts you can use to get into the whole damn thing!

Investigating Forum History (and TDO)

If your investigation leads you to forums, being able to search for historical data may become a really difficult part of your journey. I have demo'd a number of premium threat intel applications and was unable to find any that could offer the level of historical data that I needed to help further my investigation.

Many companies claimed to have data, but in reality no one did. So I ultimately had to do what Chris Robert did and built my own.

My platform is called DataViper, and building it turned out to be the cornerstone for how I was able to identify the members of The Dark Overlord (TDO). I will get much deeper into the why and how of this statement in Chapters 17 and 18. But for now, let's get back to our main story and why the ability to search forums is so crucial.

When I was researching TDO, all roads led back to the original Hell forum as the group's origins. Unfortunately, that forum shut down in 2016, and trying to find anyone with scrapes of the forum proved to be the most exhausting part of the entire investigation.

Luckily, after months and months of tireless searching, I came across one person that had the data. He has asked to remain nameless, but if he ever reads this, I want him to know just how thankful I am because his data proved to be more useful than I could ever have hoped.

In my case, all members of TDO met (and came together) in Hell. After reading through their posts and messages, the group's hierarchy became immediately clear. They seemed to all look up to someone named Cyper (no H).

By the time I had access to the Hell data, I had already figured out Cyper's other handles, so this was good affirmation that I was on the right track.

Following the closing of the Hell forum (which I believe he orchestrated), Cyper started his own forum called BlackBox under the alias Ghost. He also simultaneously started another forum called KickAss under the alias NSA.

Let me explain how I know this.

Following Breadcrumbs

EXPERTTIP: CHRIS ROBERTS

When we were looking to hack a company in the energy sector, I was wandering around looking at their third parties, their vendors, and their suppliers. I'm sitting there on one of the documents, on one of the forums, and the engineer's talking about some of the problems he's having. He puts up this link to a document. I go out to the damn document, there's a 350-page set of schematics for the entire substation, the grid, the architecture, including all the IP addresses . . .

It's the same stuff I've done whenever I've done research. It doesn't matter if it's on, obviously, the very well-known stuff, being the airplanes. I mean the airplanes was a great example, because Boeing and Airbus don't build the airplane. It's built by hundreds of other people. It's like a Lego set.

All you've gotta do is find those right people who are building the pieces you're interested in, start scraping all the intelligence from those folks, start pulling in all the data from those folks, which is kinda how I did it.

Tracing a threat actor is no different than investigating a target the same way discussed by Chris. With respect to the TDO investigation, our journey starts on 0-day (a TOR-based hacker forum), where users ze0ring and Cyper are arguing over the internal theft of the data from the Office of Personnel and Management (OPM) hack (Figure 13.1).

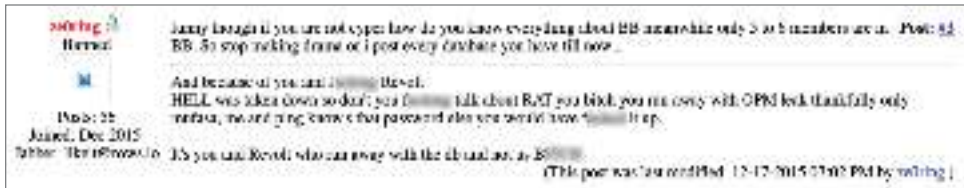


Figure 13.1

The drama is thick. But in a nutshell, ze0ring is upset because Cyper and Revolt ran off with the OPM data and caused the shutdown of the Hell forum (you can read about the details in my official report on The Dark Overlord).

Continuing to follow the drama chain, Figure 13.2 shows a message where user Photon expresses his own aggravation with BlackBox and Ghost.

For those interested, the Imgur and mega.nz links are both still active.

BlackBox was run by a user named Ghost. Among other things, Photon's screenshots of BlackBox helped confirm that Cyper and Ghost were actually the same person.



Figure 13.4

Tracing Cyper's Identity

I already knew Cyper to be the person who orchestrated the downfall of Hell (discussed in Chapter 2), so now that I was able to definitively establish Cyper as the leader of BlackBox, my interest in his identity skyrocketed. Unfortunately, BlackBox was such an exclusive forum that finding scrapes of its data proved to be impossible.

However, sometimes you find nuggets of gold in the most random places.

I have mentioned throughout this book that I created an application called DataViper, which is my own collection of credentials, hacked databases, and scraped forums. On my quest to scrape and index as much historical data as I could find, I came across a TOR social media site called Galaxy.

In fact, version 2 of Galaxy turned out to be a social media hangout for most of the Hell forum members, which included Cyper and his gang of minions.

In the following heated flame war, Cyper (aka CyPeRtRon) was defending himself against an all-out assault from two other people regarding the launch of his new forum, BlackBox, url: cyper7cybre7u57.onion.

NOTE Another hard-to-ignore indication that Cyper was the admin of BlackBox—the word Cyper is literally in the URL.

By Arsyntex

@Unknown 8698 8698, CyPeR should know, she/him are one of the skiddies from the S***box, heh, Blackbox (name of theme, very original x) & you

look forward for my Pro Forum? whats Forum? is a private network & yeah, keep dreaming, I said for Pro, not Poor ;D

By CyPeRtRoN

LOL learn what teh different between private Network and forum lame kid Hell was no private network - only a open forum ... By Arsyntex 9 days ago Soon a new HELL, only for PRO ppl, not for retarded or ignorant kids, no doxing or SQLi s***.

By CyPeRtRoN

btw - why not - when teh name is good for a forum name ... but i fear u don't know what teh name "Blackbox" stand for ...

By CyPeRtRoN

u r piss off - because you may not play to teh big boys and must play with the children all around yourself ...

By Arsyntex

hahaa I know the difference.. and I wrote "Soon a new HELL" no "Soon the same open s**** HELL"... and I'm not a kid or lame, unlike you ;D
http://matrixtxri745dfw.onion/neo/uploads/150724/MATRIXtxri745dfwONION_142610hJl_lol.png LOL

By Arsyntex

me piss off ?.. hahaa.. I "play" alone and my colleagues are clever ppl.. not idiot kids.. it's just that I like to annoy and disturb skiddies like you, I'm multitasking xD

By CyPeRtRoN

yes teh forum has rules - what want you say with this sceenshot? oh sry - u do not know this yourself ...

By CyPeRtRoN

btw - u r so smart - then certainly says you trace cookie what :D

By CyPeRtRoN

u think u hide behind tor - think again kid - not visit the wrong servers u was on my server kid ... u r not smart enough ... i have many exit nodes ... i hope for u - u don't have used one of them b**** <http://ow.ly/Q34fA>

By Arsyntex

First, you wrote "play" .. I repeat it sarcastically.. (hence the quotes). I can "play" (LOL); work alone and having colleagues, but your reasoning is so poor that you don't understand, or have colleagues means I cannot work alone ?.. and when I wro

By Arsyntex
and when I wrote "I'm multitasking", I meant that I can work while
argue with silly little girls like you xD.. the forum rules are
funny.. "then certainly says you trace cookie what :D" <-- WTF , xD

By Arsyntex
LoL ! , Now I'm scared, I disconnect better hahaha

By Sugartime
1. Rule of hidden services: Never speak of your real IP. #telnet
cyper7cyb5re7u57.onion 25 Connected to cyper7cyb5re7u57.onion. 220
ks355296.kimsufi.com ESMTP Exim 4.84 Fri, 24 Jul 2015 xx:xx:xx
+0200 #dig **ks355296.kimsufi.com 91.121.120.49**

2. Rule of hidden services: For deniability, never speak Exim ident on
your real IP. PORT STATE SERVICE VERSION 113/tcp open ident? #telnet
91.121.120.49 113 Connected to 91.121.120.49. 25, 25 : USERID :
UNIX : fail 25,25:ERROR:NO-USER

By Arsyntex
Haha xD @CyPeRtRoN
<http://freedomst2bsqtn.onion/sannucjvkdoymsycrugq/cXsgtnpE.png>

Talk about striking gold!

To add even more context behind Sugartime's post, the URL `cyper7cybre7u57.onion` is the URL of the former BlackBox forum. If this message is correct, then the IP address of BlackBox was actually exposed and was sitting on an OVH server (which is the parent host for `Kimsufi.com`), with an IP of `91.121.120.49`.

What a *massive* score!! Never underestimate the power of hackers willing to take each other down. Once they start feuding, skidz will often dox each other and do whatever they can to destroy each other.

This is a perfect example. Finding the real IP of a TOR site is virtually impossible (unless you happen to control the exit nodes), so the fact that the server was exposed like this is amazing.

Could this be the actual IP of the original BlackBox server?

Code Repositories

Code repositories, like GitHub, Bitbucket, and GitLab, can provide critical clues to finding your way into an organization, or learning about a target. In short, developers will use Git repositories to upload code. Git sites allow you to look through historical commits and will even give you an email address of the person making the commit if you know how to query it.

If you are wondering how useful this might be, let me take this time to say that I have *personally* found the following types of information stored within old Git commits:

- Private email addresses
- Hard-coded application passwords
- AWS keys
- PII and other user data
- User account passwords
- Full database archives

If you are still not convinced, the following is an actual conversation I had with NSFW, a known threat actor, regarding how he hacked a major website by finding AWS keys on the company's GitHub page:

```
BTC: i doxed the cto or whatever
BTC: cracked everything
BTC: dropbox
BTC: etc
BTC: they all had 2fa bro
BTC: but
BTC: the n****
BTC: is the most stupid idiot
BTC: cuz
BTC: github
BTC: didnt have 2fa
BTC: he put 2fa on the most useless s***
BTC: but not github
BTC: anyway
BTC: ye
BTC: got the aws creds from the github
BTC: then i priv esc'd
BTC: but
BTC: there was no data
BTC: there was nothing on the bucket
BTC: only RDS
BTC: so i had to wait for ages
BTC: till they ported over
BTC: to aws
```

SearchCode.com

SearchCode (www.searchcode.com) is a site that allows you to search code from projects on GitHub.com, BitBucket, Google Code, Gitlab, and many more sites.

Most hackers have written code, so there is a very good chance that they not only have accounts on the Git sites like GitHub and Gitlab but also have code checked in.

In addition, there is one irrefutable fact that can be applied here: *Coders will reuse their own code*. If we happen to have code written by a target, you might notice errors or interesting comments left behind. Attribution through misspellings and reused code may seem like a long shot, but it is more common than you may think.

NOTE Even better, when threat actors first apply for access to hacker forums, they almost always include sample code in their application. Being able to go back and collect these code samples can be extremely useful.

Back to SearchCode.com, the results are not great, but I have not found any competing sites that are any better. In my opinion, the main issue I have with the site is that you can't search for full strings.

For example, searching for “really long string” will return results that match *any* of the words really, long, or string, which can lead to a large amount of useless results.

That being said, it's better than nothing, and if your results are not found, the app gives you quick access to perform a direct search of your string on the major Git sites.

Searching for Code

The interface on SearchCode.com is pretty self-explanatory. Enter your query in the search box (Figure 13.5) and click Search.



Figure 13.5

For our test search, let's try the string “\x48\x31\xc0\x5e\x68”, which will yield quite a few results. The left part of the results window in Figure 13.6 is what I find most useful about this site.

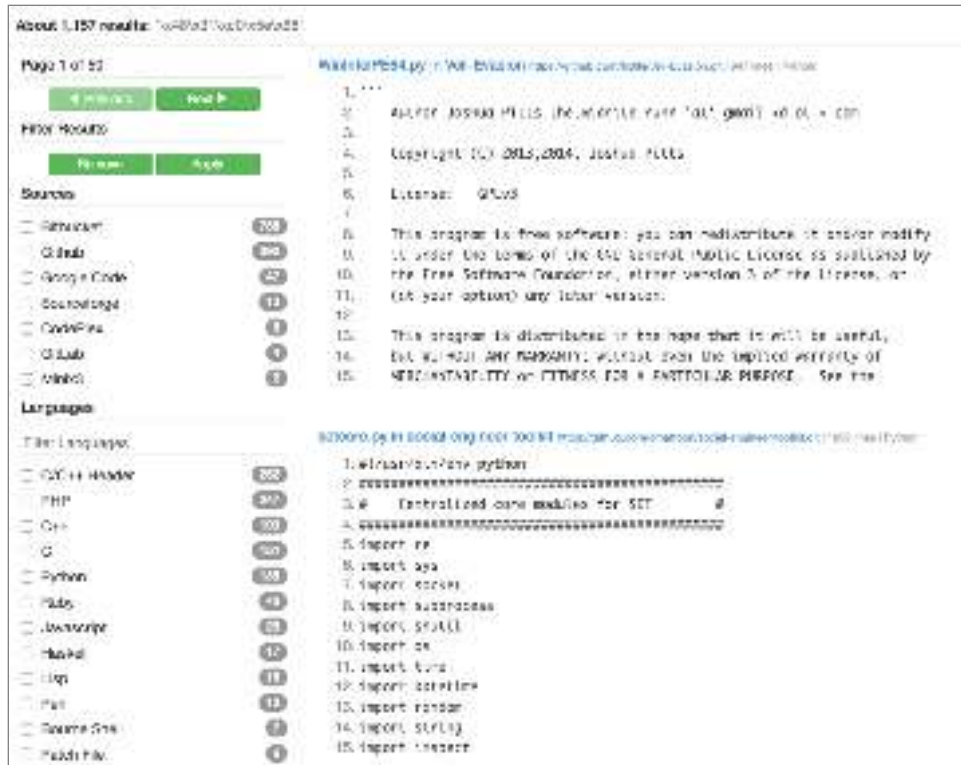


Figure 13.6

Our search yielded too many results to be useful, so using the filter panel can help refine our search results. This is a huge time-saver and will allow you to quickly explore code from different sites or languages by simply selecting the appropriate filter.

False Negatives

One important item to note when using this site is the number of false negatives you will receive in your results. What I mean by that is your search may often return “no results found,” or even worse, some unrelated or incorrect results on search terms. For example, let’s try searching for “c3nt3rx” (which is an alias of a threat actor from the KickAss forum).

The results page (Figure 13.7) shows that we have no matches.

As mentioned earlier, when no results are found, SearchCode.com provides quick links to search for your term on several different sites, including GitHub. Clicking the link for GitHub takes us right to that site and automatically performs the same search by including the term in a URL query.



Figure 13.7

As you can see in Figure 13.8, GitHub has 15 matches for the term “c3nt3rx.” Lucky for us, he was the primary developer of the KickAss Framework, a hacking toolkit maintained by some of the KickAss forum members (including Cyper).



Figure 13.8

The big takeaway here is that no tool is ever perfect, and you should always check your results with multiple sources. When using tools that aggregate results from multiple sites, it is also worth checking the original site to ensure that your results are accurate and consistent between the aggregating site and the original site.

Now that we have a new lead to explore apropos the KickAss framework, let's see if we can learn anything more from the GitHub page.

Gitrob

Gitrob is an open-source tool used to search for potentially sensitive files or information stored in public GitHub repositories. Gitrob is useful to researchers and developers alike because it will scan a repository and flag information that can be potentially harmful or private.

Gitrob works by cloning public repositories and looking through the commit histories. It flags potentially sensitive files and presents its findings via a web interface for further analysis. Results can also be exported to a JSON file.

Running Gitrob against c3nt3rx's repo will give us the following results:

```
> gitrob -bind-address 0.0.0.0 c3nt3rx

  ____  _  ( )  /_____/  / /
 / _  ` / /  _/  _/  _/  _/  _/
 \_, / _/ \_/_/  _/  _/  _/  _/
 /___/ by @michenriksen

gitrob v2.0.0-beta started at 2019-07-15T03:34:29Z
Loaded 91 signatures

Web interface available at http://0.0.0.0:9393
Gathering targets...

Retrieved 7 repositories from c3nt3rx
Analyzing 7 repositories...

Findings.....: 0
Files.....: 77
Commits.....: 63
Repositories: 7
Targets.....: 1
```

WARNING These results may appear misleading. Even though the results show 77 files and 63 commits, if Gitrob does not detect anything interesting, the web interface will show 0 results.

In this particular case, since there are no results that Gitrob considered “interesting,” the web interface will show 0 results.

To show a contrasting example with better results, I’ve run Gitrob against the repository of michenriksen (the developer of Gitrob):

```
gitrob -bind-address 0.0.0.0 michenriksen

  ____  _  ( )  /_____/  / /
 / _  ` / /  _/  _/  _/  _/  _/
 \_, / _/ \_/_/  _/  _/  _/  _/
 /___/ by @michenriksen

gitrob v2.0.0-beta started at 2019-07-15T03:31:42Z
Loaded 91 signatures
```

```
Web interface available at http://0.0.0.0:9393

Gathering targets...
  Retrieved 20 repositories from michenriksen

Analyzing 20 repositories...

MODIFY: Contains word: credential
  Path.....: credentials.json
  Repo.....: michenriksen/searchpass
  Message....: Update passwords
  Author.....: Michael Henriksen <michenriksen@neomailbox.ch>
  File URL...: https://github.com/michenriksen/searchpass/blob/
a245aee..[truncated]
  Commit URL.: https://github.com/michenriksen/searchpass/commit/
a245ae..[truncated]

MODIFY: Contains word: credential
  Path.....: credentials.json
  Repo.....: michenriksen/searchpass
  Message....: Update passwords
  Author.....: Michael Henriksen <michenriksen@neomailbox.ch>
  File URL...: https://github.com/michenriksen/searchpass/blob/
ff908..
[truncated]
  Commit URL.: https://github.com/michenriksen/searchpass/commit/
ff9085c..[truncated]

Findings.....: 2
Files.....: 539
Commits.....: 225
Repositories: 20
Targets.....: 1

Press Ctrl+C to stop web server and exit.
```

The difference in results is immediately visible. Gitrob provides a description of what “interesting” items are detected, and their location. In the case of this repository, the results show that certain files contain the word “credential.”

Now when we access the web server, we can see a more detailed list of the discovered items. Figure 13.9 shows each of the discovered items along with direct links to view them.

Gitrob is programmed to search for things like SSL keys, stored credentials, and a slew of other information. If you come across a data-rich repository, this screen will be lit up with information.

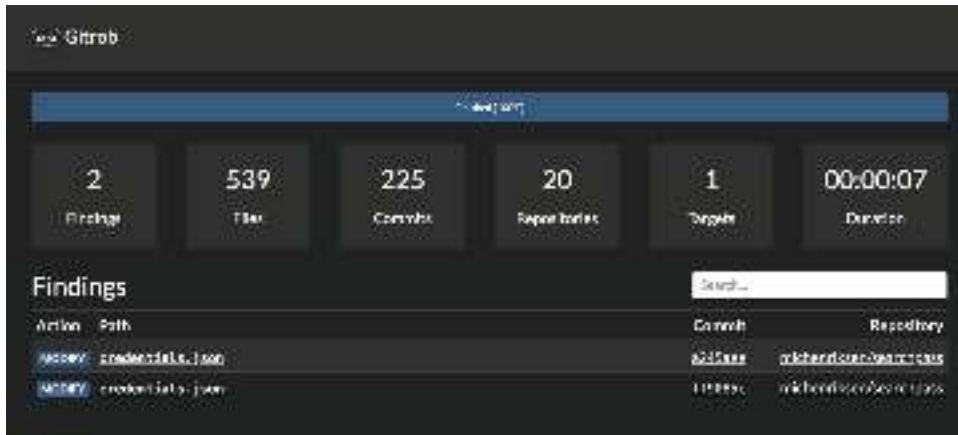


Figure 13.9

Git Commit Logs

Another approach to gathering information can be as simple as looking at the repository's Git commit logs. The Git logs show the time and date of every commit, including information on the person who committed the changes. Since each commit *must* have an associated email address, the logs can reveal new information about a target, such as new aliases or email addresses.

To illustrate my point, let's investigate c3nt3rx's repository for the "KickAss Framework." The framework was openly discussed on the KickAss forum and can be directly tied to several of the forum's key members, including NSA, the site's admin (who is actually our buddy Cyper).

For those playing along at home, the URL for the framework is <https://github.com/c3nt3rX/kaf>.

After checking out the repository, you can see the commit logs by typing `git log`:

```
[root@OSINT] > git log

commit 9a8d392f4265f9fafec854d06bcc86608c393b3a
Author: NSA <nightssquare@sigaint.org>
Date: Thu Jun 16 22:11:16 2016 +0200

    changes

commit c9282534f031f066450197f31ef985d07661daa7
Author: NSA <nightssquare@sigaint.org>
Date: Thu Jun 16 22:09:33 2016 +0200

    some changes and new scripts
```

```
commit 20fa61ff8113dd34e2dd6a2485b9654d6e09459a
Author: NSA <nightsquare@sigaint.org>
Date:   Fri May 27 01:18:05 2016 +0200
```

new banner

```
commit af98fc17cec67f8a3085f374161cec93e15cd177
Author: NSA <nightsquare@sigaint.org>
Date:   Wed May 25 19:01:38 2016 +0200
```

new readme

```
commit fd594e4bbc70e184005a7a3931a02aa7d3613b5a
Author: c3nt3rX <centerx@hotmail.gr>
Date:   Sat May 21 02:53:06 2016 +0300
```

Update kaf.py

```
commit c06b68662dalb8690963a47930d24f04e6a75028
Author: c3nt3rX <centerx@hotmail.gr>
:...skipping...
commit 6fbe9999d2fb68fb0954866328ad63505f4a06a5
Author: NSA <nightsquare@sigaint.org>
Date:   Thu Jun 30 08:06:54 2016 +0200
```

change donate adresse

Looking at the results, we can see two different authors: NSA and c3nt3rx. Each of the authors has its own associated email address—we now have an email address for c3nt3rx, and a really interesting email address for NSA (Cyper)—nightsquare@sigaint.org.

NOTE “NightSquare” is a really interesting email address because the first two letters in NSA can be mapped to Night and Square. I don’t know if this is just a coincidence, or if there is more meaning behind the name NSA. If so, what does the “A” stand for?

The next section will show that Cyper lives in (or near) Austria, so maybe the name is a reference to Austria’s Night Square?

I was never able to figure that out, so if anyone reading this has any ideas please send me a message.

Wiki Sites

To me, there is nothing more satisfying than the feeling of finding that “jackpot” piece of evidence during an investigation. This feeling is multiplied if you also get to end it with the phrase “what an idiot” [SMH].

This brings us back to the story of our friend Cyper (aka CyPeRtRoN, aka Ghost, aka NSA).

One of Cyper's more recognizable traits is the way he *always* talks about his affiliation with Hackweiser (a hacking group from the late '90s). He would actually never shut up about it, which made tracking him across accounts easier. Once you know what to look for, or have some sort of key identifiable information to go on, the pieces will start to fall into place.

That being said, on July 24, 2015, the following discussion took place on the Galaxy 2, a TOR-based social media site, where users Cyper and Arsyntex were involved in another heated discussion:

July 24, 2015

By CyPeRtRoN

for all other - don't be paranoid - don't f*** with me and all is good ...

By Arsyntex

Beware with @CyPeRtRoN is a pro cookie tracer.. It is one of the b***** of the NSA and GCHQ xD

By CyPeRtRoN

LOL nice thx for the promo - u don't know who am i - u think u release 2 exp and now u ar teh man Jogesh

By CyPeRtRoN

something to read for u <https://en.wikipedia.org/wiki/Hackweiser>

By Arsyntex

Members included; R4ncid, Bighawk, [P]hoenix, Immortal, RaFa, Squirrlman, PhonE_TonE, odin, x[beast]x, Phiz, @CyPeRtRoN and Jak-away (AKA Hackah Jak). **hahahaah** (¬¬¬)

There are two interesting and important takeaways from this discussion: the Wikipedia URL for Hackweiser, and Arsyntex's reaction to Cyper posting the URL.

Up until this moment, I never considered Wikipedia (or any other public wiki) to be a possible source of credible intelligence.

I was wrong.

Wikipedia

Wikipedia, despite its general lack of credibility as a trusted source of information, does one thing really well: it keeps an accurate and public record of every change made to its pages (requested or permanent).

The current Wikipedia site for Hackweiser (<https://en.wikipedia.org/w/index.php?title=Hackweiser>) lists the following members: R4ncid, Bighawk, [P]hoenix, Immortal, RaFa, Squirrlman, odin, x[beast]x, Phiz, and Jak-away(AKA Hackah Jak).

As you may have noticed, there is a key member missing from this forever immortalized list: the great CyPeRtRoN.

But wait . . . We just saw a post from Galaxy 2 where Cyper was clearly listed as a member of Hackweiser. He was even bragging about it.

Remember, in the eyes of a young and aspiring hacker, *Vanity will always trump OPSEC*.

To prove my point, we can consult the View History tab located on every Wikipedia page (shown in Figure 13.10).



Figure 13.10

On the View History page for Hackweiser (Figure 13.11), we see the full list of changes made to the page (this list includes both permanent and removed changes).



Figure 13.11

Going back to our conversation from Galaxy 2, the message from Cyper was dated June 2015. Notice in Figure 13.11 that there is an entry for May 2015? To understand the full impact of the May 2015 change, let's first look at the preceding entry for November 2014 (which you can view directly at <https://en.wikipedia.org/w/index.php?title=Hackweiser&oldid=635593910>).

According to this page from November 2014, the members of Hackweiser are R4ncid, Bighawk, [P]hoenix, Immortal, RaFa, Squirrlman, Phone_TonE, odin, x[beast]x, Phiz, and Jak-away(AKA Hackah Jak)—the exact same list that is on the current page.

Wikipedia also has a very handy Compare Selected Revisions button (i.e., a diff tool). Each revision to the page shown in Figure 13.11 has a radio button next to it. Selecting the radio button for the May 2015 revision, then clicking Compare Selected Revisions, will take us to a diff page showing the differences between the two entries (Figure 13.12).



Figure 13.12

You can also view this diff page directly via this link: <https://en.wikipedia.org/w/index.php?title=Hackweiser&diff=662753224&oldid=635593910>.

As you have probably already guessed, the member list from May 2015 contained one extra special addition: **CyPeRtRoN**.

Could our friend Cyper be so vain as to purposefully edit his own name into the Wikipedia page? And more importantly, would he be willing to take the risk of making this edit from a non-VPN IP address in order to add legitimacy to his edit?

In case you missed it the first time: *Vanity will always trump OPSEC*.

A quick search on whatismyipaddress.com (Figure 13.13) gives us the approximate location of 62.93.70.34, the IP address used to make the change to the Wikipedia page.

Could Cyper have been so vain as to make this Wikipedia edit directly from his house? I can't legally say for certain, but that is a pretty solid lead!



IP Details for 62.93.70.34

This information should not be used for emergency purposes. Trying to find someone's exact physical address, or other purposes that would require 100% accuracy.

62.93.70.34 [Lookup IP Address](#)

Details for 62.93.70.34

IP: 62.93.70.34
Decimal: 1046300194
Hostname: monitor.jm-data.at
ASN: 25447
ISP: JM-DATA GmbH
Organization: JM-DATA GmbH
Services: None detected
Type: [Broadband](#)
Assignment: [Static IP](#)
Blocklist: [Click to Check Blocklist Status](#)
Continent: Europe
Country: Austria 🇦🇹
State/Region: Tyrol
City: Hopfgarten in Denzlingen
Latitude: 46.6247 (46° 55' 28.92" N)
Longitude: 12.4058 (12° 29' 44.88" E)
Postal Code: 9961

Figure 13.13

NOTE You can find additional details placing Cyper in or around Austria in the official investigation report on The Dark Overlord.

Summary

This chapter was focused on uncovering hidden information throughout various scattered corners of the Internet. This includes using tools like theHarvester for wide-range information gathering, and more focused tools like Gitrob for gathering information for GitHub.

The details and examples provided throughout this chapter should make it clear that even the most random (and obscure) places like forums, paste sites, code repositories, and wiki pages can offer very unexpected results, and sometimes ground-breaking clues in an investigation.

In the next chapter, we will expand our search even further to look for information from publicly accessible databases like MongoDB and Elasticsearch.

Publicly Accessible Data Storage

Discovering data breaches (or data leaks) is never an easy thing. What I mean by that is figuring out how to deal with the situation once you have discovered the breach or leak is, in my experience, the most difficult part of the entire process.

This chapter will focus on both sides of that story. We will look at methods for discovering data in several common NoSQL databases like MongoDB and Elasticsearch, as well as data located on publicly accessible cloud storage platforms such as Amazon buckets and Digital Ocean spaces. I will also reveal some of my personal discoveries and experiences surrounding those discoveries.

But first, a word from Bob Diachenko, a man who has personally uncovered more high-profile data leaks than anyone else I know.

EXPERTTIP: BOB DIACHENKO

There is no stable pattern that I will use [to find data] every time, because I change [the methods] based on the results that I receive. And I also deliberately try to minimize the automating of search results. So when I look through reports, I always do that manually. I do this because in the past, I have overlooked important information and made a lot of mistakes when I rely on automated scripts.

When I started reviewing the results, I was surprised to see that many interesting items were overlooked. So then I started to go through the results myself, which is how I came up with my current process.

I reasonably avoid using any sophisticated or intrusive techniques, or anything that might associate me with being a “blackhat” hacker. The tools I use are always public. My message here is that if I am able to find your data, then anybody in the world can do this. So when I find something, it is available to anyone on the Internet.

But it was not until I started doing responsible disclosure that I started to receive blowback from companies. Especially from companies that are really hard to contact.

I think it may be because I had no profile at the time—I mean, I was a completely random person to these companies. They were suspicious, and that was a real hard challenge to overcome.

I have had moments when I almost quit doing this. A couple of companies tried to sue. I don’t know if that was perhaps because I was acting as part of a company, so maybe they were hoping that they would sue the company and make some money? I honestly don’t know.

I eventually realized the value that I bring to the community, no matter how hard the pressure is. I also reassessed the way I approach companies, and also the way I handle the data. These companies that I notify are embarrassed by what has happened, and in many cases are legally accountable for leaving the data exposed, so I now follow a different process, which seems to be working.

In these types of situations, you need to be really careful with the data that you’re looking at. Be prepared to explain how you handled the data, why you made certain decisions, and be ready to make assurances that you are the only one who is looking at your copy of the data (I say *your* copy because since the data is public many people can be seeing it).

Unfortunately, most of the companies never ask this of me. They just take the information and never ask me to sign an NDA, or even try to how I handle the data or where I store it. That’s really interesting.

The Exactis Leak and Shodan

Around July 2018, I uncovered a data leak by Exactis—this was my first major public Elasticsearch database discovery. Exactis was a marketing firm (i.e., data broker) that made money selling data to other companies. The public database contained highly personal information on more than 200 million people. The data included names, email addresses, phone numbers, addresses, gender, religious beliefs, political preferences, pet information, household sizes, and a slew of other lifestyle-specific information such as whether a person smokes, if they scuba dive, wear plus-size clothing, etc.

The important thing to note about Elasticsearch databases is that they do not have user permissions in the same way as most databases. By default, you do not need to authenticate to the data using a username and password. Roles and permissions can be configured but by default, there is nothing.

That means that anyone trying to access your Elasticsearch database via IP address on port 9200 will have full CRUD (create, read, update, delete) rights to your data simply by querying the URL. The way to protect your ES cluster is to put everything behind a firewall so that the IPs are not publicly exposed.

Data Attribution

Discovering the owner of an exposed data set is usually the most difficult part of the investigation because all you have to go on is an IP address, and a hosting company will never give up the owner of that address without a formal legal request.

While I typically contact the organization long before I speak with anyone else, this case was different. Trying to identify the owner of an IP address can be one of the most grueling and exhausting tasks you will have to undertake, so I enlisted a few people to help.

It took me a solid month to figure out who actually owned the Exactis data. After several weeks of not being able to find the owner, I reached out to law enforcement and reporters that could potentially help.

By the time we figured out that Exactis was the owner of the exposed servers, I had already been working closely with Andy Greenberg at *Wired*. When we did finally figure it out, I contacted Exactis immediately and advised they take the data offline because it was completely wide open.

I explained that I had already notified law enforcement because the data appeared to contain information on almost every U.S. citizen, and also that I was working with *Wired* during my investigation. I strongly advised they speak with Andy to get their side of the story on record to help with the damage control.

What happened to Exactis following the publishing of the *Wired* story is not something I could have predicted, and something I still feel bad about to this day. The company went out of business and the owner, Steve Hardigree, is currently facing several class-action lawsuits. One night I even received a text message from Steve asking me why I ruined his life.

I'm not going to lie, that sucks because I know there is some truth in that. On the other hand, he is the CEO of a company that left all this data exposed. At some point, this was his responsibility. And frankly, given how many researchers look for open data storage, it was only a matter of time before someone else would have found the data.

According to the most recent story by *Wired*, the CEO states that the data was only left open "for a matter of days." I have a different perspective on this issue based on how long it took me to find the data.

That being said, my "word" means nothing. There will come a time in every investigator's career when it will be necessary to back up your words with

facts, and it was this situation that taught me several new amazing and useful Shodan techniques.

Shodan's Command-Line Options

The majority of researchers use Shodan's web interface, but many people may not realize (as I didn't) all of the extended capabilities available through Shodan's command-line tools.

One of the great benefits of Shodan's command-line tools is the ability to download all of the results of a particular database type.

For example, one use would be to have Shodan give us all the IPs of publicly accessible Elasticsearch databases. Doing this requires two separate commands. The first to download the data, and the second to parse it into a usable format. This was the exact process I used to uncover the Exactis data leak.

We can first tell the Shodan tool to download all IPs running Elasticsearch using the following syntax:

```
Shodan download --limit (number of results) (your filename) (query)
```

For a full download of all public Elasticsearch databases, I set the limit to 50,000 entries. The web UI shows roughly 30,000 entries, so I intentionally set this number higher to make sure everything was included:

```
root@osint: > shodan download --limit 50000 elasticdata product:Elastic
```

```
Search query:           product:Elastic
Total number of results: 28289
Query credits left:     xxxx
Output file:            elasticdata.json.gz
```

Now that we have the downloaded results, we can use the `parse` command to extract the relevant data that we need. In this case, I typically just extract the IP and port number using the following command:

```
shodan parse -fields ip_str,port --separator , elasticdata.json.gz
```

That will create a parsed file of just IP addresses and port numbers. Later in this chapter we will use that information to automate the process of identifying servers that may be leaking sensitive information.

Querying Historical Data

Earlier in this section I mentioned that the Exactis scenario taught me something really useful about Shodan's command-line tools. In addition to being able to easily query and download raw data, Shodan's tools also allow you to search historical information about IP addresses. This means that over any given time, Shodan will not only identify which ports are open on an IP address, but it will

also keep a historical record of that data that you can query. Let's apply this to the Exactis IPs and see what we can find.

Exactis had three IP addresses associated with its Elasticsearch cluster:

- 172.106.108.69
- 172.106.108.73
- 172.106.108.77

As part of an investigation, you may (and most likely will) be asked to provide some level of evidence (or testimony) regarding the length of time that an IP address or data set was online. In a follow-up interview with *Wired* magazine, CEO of Exactis Steve Hardigree "insists the data was left exposed only for a matter of days."

Let's see what Shodan tells us.

The following syntax will query historical information on an IP address:

```
shodan host --history -S --format pretty (ip address)
```

Running this against one of the Exactis IPs will return the following information:

```
[ root@OSINT ] > shodan host --history -S --format pretty 172.106.108.77
```

```
172.106.108.77
City:                Ashburn
Country:             United States
Organization:        Psychz Networks Ashburn
Updated:             2019-04-11T06:43:30.053445
Number of open ports: 7
Vulnerabilities:      CVE-2018-15919 CVE-2018-15473 CVE-2017-15906
```

```
Ports:
  25/tcp              (2016-09-02)
  25/tcp              (2016-08-30)
  25/tcp              (2016-06-18)
  25/tcp              (2016-06-16)
  80/tcp Apache httpd (2.2.15)      (2016-10-14)
  2020/tcp OpenSSH (7.4)          (2019-04-11)
  3306/tcp MySQL      (2016-06-02)
  5601/tcp            (2018-06-01)
  5601/tcp
[results truncated]
9200/tcp Elastic (6.2.4)          (2018-06-04)
  9200/tcp Elastic (6.2.4)          (2018-06-02)
  9200/tcp Elastic (6.2.4)          (2018-05-29)
  9200/tcp Elastic (6.2.4)          (2018-05-25)
  9200/tcp Elastic (6.2.2)          (2018-05-07)
  9200/tcp Elastic (6.2.2)          (2018-04-27)
  9200/tcp Elastic (6.2.2)          (2018-04-03)
  9200/tcp Elastic (6.2.2)          (2018-03-31)
  9200/tcp Elastic (6.2.2)          (2018-03-20)
```

9200/tcp Elastic (6.2.2)	(2018-03-18)
9200/tcp Elastic (6.2.2)	(2018-03-11)
9200/tcp Elastic (6.1.1)	(2018-02-19)
9200/tcp Elastic (6.1.1)	(2018-02-15)
9200/tcp Elastic (6.1.1)	(2018-02-15)
9200/tcp Elastic (6.1.1)	(2018-02-10)
9200/tcp Elastic (6.1.1)	(2018-01-21)
9200/tcp Elastic (6.1.1)	(2018-01-03)
9200/tcp Elastic (6.0.0)	(2017-12-14)
9200/tcp Elastic (6.0.0)	(2017-12-07)
9200/tcp Elastic (6.0.0)	(2017-11-24)

What we can see from these results is that port 9200 (the typical port for an Elasticsearch server) was first detected by Shodan on November 24, 2017, and remained open until June 4, 2018 (which was around the time I contacted them and the story broke).

This means that according to Shodan’s historical information, the servers were open for a full seven months before I discovered them. This is why I have trouble accepting the Exactis CEO’s statement that the servers were only online for “a matter of days.”

That bit of information at least made me feel better about the discovery, and the ultimate outcome of the organization. I can completely understand the CEO’s anger and resentment toward me. However, a seven-month period is a completely different situation. Given how many other companies, researchers, and criminals scrape public data sets, it is highly probable that I am not the only person who found this data, but perhaps was the only person able to figure out who owned it.

NOTE I guess in the end the damage to his life wasn’t too bad because Steve is back to selling data with his new company, BrightSpeed (www.brightspeed.com).

I hope my account not only helps shed light on the events surrounding the discovery and announcement of the Exactis incident, but also helps give more context around the types of situations and repercussions that investigators may face during their career.

CloudStorageFinder

CloudStorageFinder (CSF) is an open-source tool written by Robin Wood that does exactly what the name implies—it finds publicly exposed cloud storage buckets. A number of cloud storage finder tools are available, but I chose CSF because of its capability to search more than just AWS.

CSF is able to search through publicly accessible Amazon S3 buckets, Digital Ocean spaces, and SpiderOak shared folders. CloudStorageFinder is available at <https://github.com/digininja/CloudStorageFinder>.

CSF works by bruteforcing public URLs, so the better your wordlist, the more results you will have. Refer back to Chapter 2 on methods to create great wordlists for bruteforcing.

CSF has three key tools: `bucket_finder.rb`, `space_finder.rb`, and `spider_finder.rb`. As you can probably imagine, `bucket_finder.rb` is used to look for Amazon S3 buckets, `space_finder.rb` is used for finding Digital Ocean spaces, and `spider_finder.rb` is used for finding SpiderOak shared folders.

The options and parameters for all three programs are very similar, the main difference being the region parameters. In each case, public storage is located in different regions, so this parameter may change depending on the different services you are looking for.

The parameters of the CSF S3 bucket finder include:

```
--help, -h: show help
--download, -d: download the files
--log-file, -l: filename to log output to
--region, -r: the region to use, options are:
    us - US Standard
    ie - Ireland
    nc - Northern California
    si - Singapore
    to - Tokyo
-v: verbose

wordlist: the wordlist to use
```

Amazon S3

Assuming you have your brute force wordlist ready to go, running CSF is pretty straightforward. For our initial scans, we will run `bucket_finder` against the U.S. region and download everything we come across using `-d`:

```
[ root@osint ] > ./bucket_finder.rb -r us -d wordlist.txt -l logfile.txt
```

The output of my initial scans looked something like this:

```
Bucket iis redirects to: iis.s3.amazonaws.com
Bucket does not exist: endofspecialwords
Bucket does not exist: Aarhus
Bucket found but access denied: Aaron
Bucket does not exist: Ababa
Bucket found but access denied: aback
```

```
Bucket does not exist: abaft
Bucket does not exist: abandoned
Bucket does not exist: abandoning
Bucket does not exist: abandonment
Bucket does not exist: abandons
Bucket found but access denied: abase
```

One thing CSF does not do is allow for bucket authentication of the use of S3 API keys. This will be covered later in this chapter using the NoScape tool.

Digital Ocean Spaces

Digital Ocean is my cloud provider of choice. Digital Ocean recently released a “spaces” feature, which is another type of public bucket, similar to Amazon S3. This is great because I always like looking through new services because there is a great chance for misconfiguration right out of the box.

NOTE When you sign up for a Digital Ocean space, the signup process asks whether you want to have public access or restrict access to only users with correct keys.

Amazon’s initial setup and configuration is *significantly* more complex, so I personally don’t think there will be much room for error here, unless someone accidentally creates a public space and forgets about it. Nevertheless, we should still try.

The main difference in running CSF’s `space_finder` is the slight change in parameters. The different parameters include:

```
Usage: space_finder [OPTION] ... wordlist
-h, --help: show help
-d, --download: download the files
-l, --log-file: filename to log output to
-h, --hide-private: hide private spaces, just show public ones
-n, --hide-not-found: hide missing spaces
-r, --region: the region to check, options are:
    all - All regions
    nyc - New York
    ams - Amsterdam
    sgp - Singapore
-v: verbose

wordlist: the wordlist to use
```

For our test run, we will try running `space_finder` against all regions using the `-r all` parameter. Everything else will be the same. Running the tool against “all” regions means that you will be sending three different requests, one for each region:

```
[ root@osint ] > ./space_finder.rb -r all -d wordlist.txt -l logfile.txt
```

The output will look something like this:

```
Space does not exist in region ams3: Backup
Space does not exist in region nyc3: Backup
Space does not exist in region sgp1: Backup
Space does not exist in region ams3: warez
Space does not exist in region nyc3: warez
Space does not exist in region sgp1: warez
Space does not exist in region ams3: pr0n
Space does not exist in region nyc3: pr0n
Space does not exist in region sgp1: pr0n
Space does not exist in region ams3: porn
Space does not exist in region nyc3: porn
Space does not exist in region sgp1: porn
Space does not exist in region ams3: Scripts
Space does not exist in region nyc3: Scripts
Space does not exist in region sgp1: Scripts
Space does not exist in region ams3: IISHelp
Space does not exist in region nyc3: IISHelp
Space does not exist in region sgp1: IISHelp
Space found in region nyc3: vinnytroia
(https://vinnytroia.nyc3.digitaloceanspaces.com )
<Private> https://vinnytroia.nyc3.digitaloceanspaces.com/test/
```

If you get lucky and find a public space, it will look like the preceding output for my personal space.

For now, let's move on to NoSQL databases and the wonders of what can be found there.

NoSQL Databases

Traditional relational databases had scalability issues and schema flexibility, which led the way to NoSQL database types. These days, most modern web applications use some sort of NoSQL database. In a NoSQL database, everything is document-based, and not stored in traditional “tables.”

In addition to their speed and flexibility, most of these database systems also require security configurations above and beyond that of traditional SQL databases. For example, most Elasticsearch databases are publicly accessible by default, unless you put some sort of firewall in front of them or block access to port 9200.

This section will look at ways to find publicly accessible data stored on several of the more popular NoSQL databases including MongoDB, Elasticsearch, and CassandraDB.

MongoDB

MongoDB is one of the most popular NoSQL database types. MongoDB stores data in flexible JSON-like documents, which means it does not use the traditional SQL-like table structure found in many applications. You can query MongoDB databases in a number of ways, including the Mongo command-line tools and GUI applications like RobotMongo.

EXPERTTIP: BOB DIACHENKO

For Mongo, it's interesting because I think they made a mistake in the past that they are paying for now. In one of the earlier versions, they decided to leave the default configuration without any passwords or authorization—and that version was quite popular. Still, many companies and administrators around the world are using that obsolete version of MongoDB, and have not updated their software to a newer version where those default credentials are not allowed. The full “credentials” are just no password at all, so it's not like you even have to try to bruteforce. You just don't use any password and you are able to log in.

Robot 3T

Robot 3T is a free GUI tool used for accessing and working with MongoDB databases. I personally find the Mongo command-line utilities to be cumbersome and temperamental. Robot 3T also has a premium version called Studio 3T, which includes the ability to import and export from a Mongo database.

Jumping right in, if you have no connections saved in Robot 3T, the first screen will prompt you to enter information about your connection (Figure 14.1).

This section will show screens taken from an actual live MongoDB server. Since I don't know who owns the data (and because it's not mine), all pertinent information will be hidden.

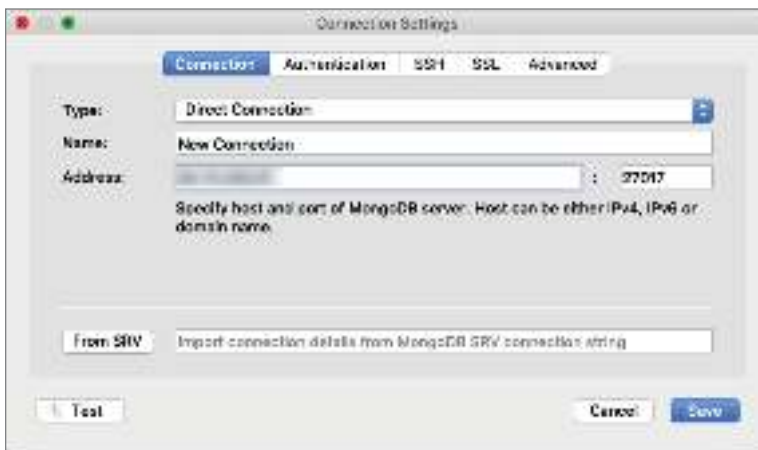


Figure 14.1

After you enter the target server's IP address and click Save, you will be taken to the list of MongoDB connections (shown in Figure 14.2).

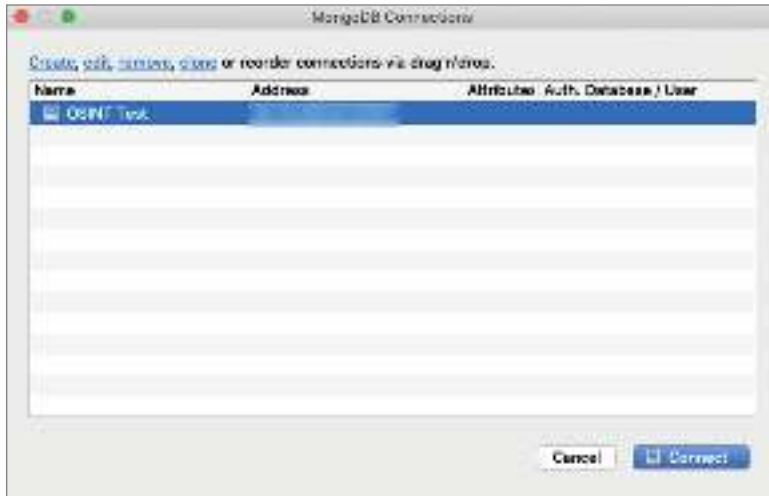


Figure 14.2

Clicking the Connect button will connect you to your MongoDB server. The first thing you will notice in the left column is a tree menu with a list of available databases and collections. Figure 14.3 shows the databases available on the test server.

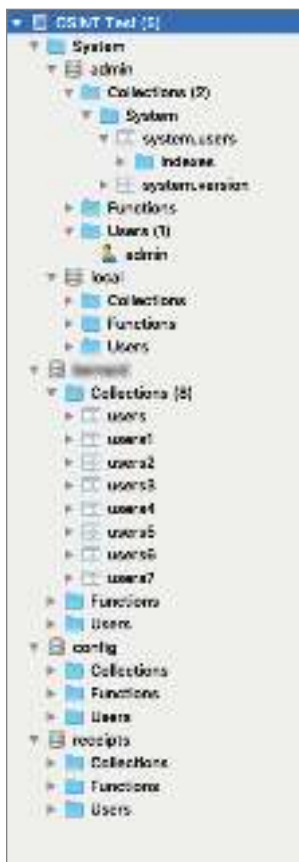


Figure 14.3

For easy reference, I will refer to the blurred database name in Figure 14.3 as PrivateDB.

Expanding each of the databases will show the list of collections (similar to tables) available. Figure 14.4 shows the list of expanded collections for each database.

Since PrivateDB looks the most interesting, let's continue to explore this database. Within PrivateDB, we can see eight different collections. Clicking the first "users" collection will expand the application window and show us a list of documents available within that collection (Figure 14.5).

**Figure 14.4**

We can see from the list view that the majority of documents have 11 (or 12) fields. To view the different fields within any of the documents, click the arrow and it will expand your view to match Figure 14.6.

This view also allows full access to create, modify, or delete any of the fields. Since this is not our data, this is as far as we will go. However, modifying the data is as easy as right-clicking the field you want and using the menu (Figure 14.7).

WARNING The free version of Robot 3T does not allow you to export data. If you want to capture the entire database, you will need to either use the command-line tools or purchase a license for the full version of the tool.

Robot 3T is great for quickly browsing individual MongoDB databases, but trying to quickly view and process tens of thousands of IPs using this method will be impossible. Later in this chapter, we will look at a tool called NoScrape, which will help with that function. Before that, let's take a quick look at the Mongo command-line tools.

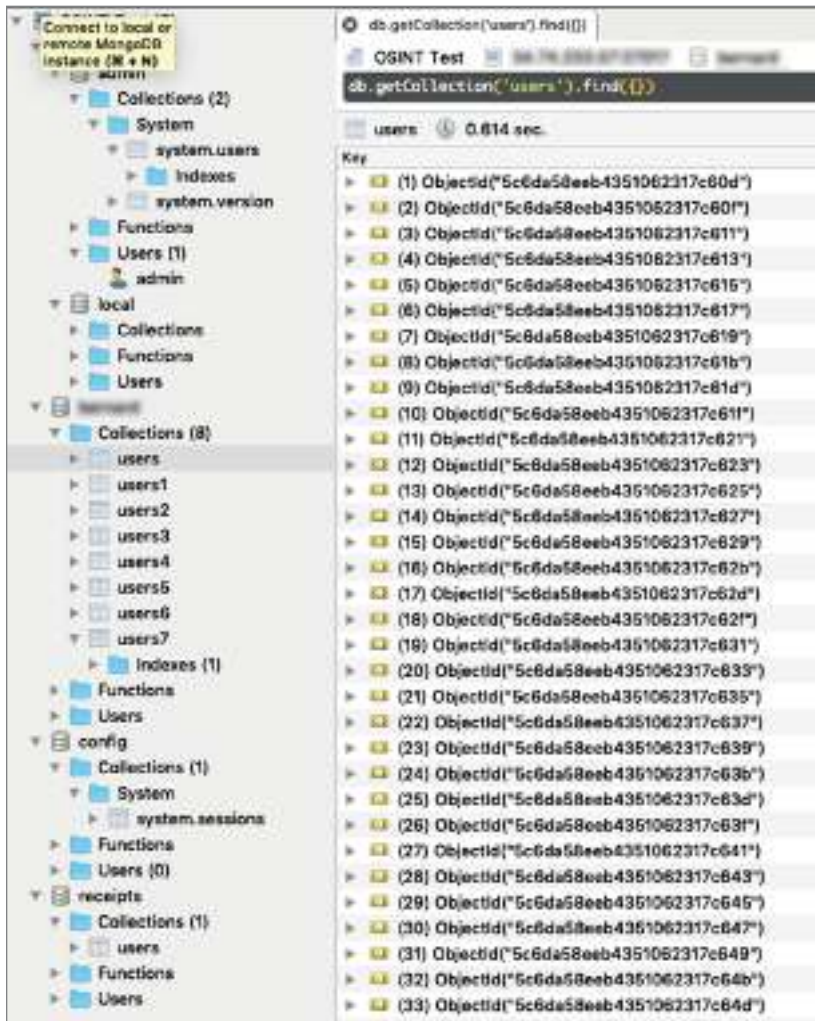


Figure 14.5

Mongo Command-Line Tools

Mongo's command-line tools are equally useful for browsing and dumping a Mongo database (although dumping is somewhat more difficult than browsing). This section is only meant to provide a very brief look at Mongo's default tools since our main focus on dumping will be later in the chapter with NoScrape.

Once the tools are installed, connect to a Mongo database by typing `Mongo` followed by the IP address.

```
[ root@scraper1 ~ ] > mongo 0.0.0.0
MongoDB shell version v3.4.20
connecting to: mongodb://0.0.0.0:27017/test
```

```

MongoDB server version: 4.0.9
WARNING: shell and server versions do not match
Server has startup warnings:
2019-04-17T00:03:45.291+0000 I STORAGE [initandlisten]
2019-04-17T00:03:45.291+0000 I STORAGE [initandlisten]
** WARNING: Using the XFS filesystem is strongly recommended
with the WiredTiger storage engine
2019-04-17T00:03:45.291+0000 I STORAGE [initandlisten]
**See http://dochub.mongodb.org/core/prodnotes-filesystem
2019-04-17T00:03:48.012+0000 I CONTROL [initandlisten]
2019-04-17T00:03:48.012+0000 I CONTROL [initandlisten]
** WARNING: Access control is not enabled for the database.
2019-04-17T00:03:48.012+0000 I CONTROL [initandlisten]
** Read and write access to data and configuration is unrestricted.
2019-04-17T00:03:48.012+0000 I CONTROL [initandlisten]

```

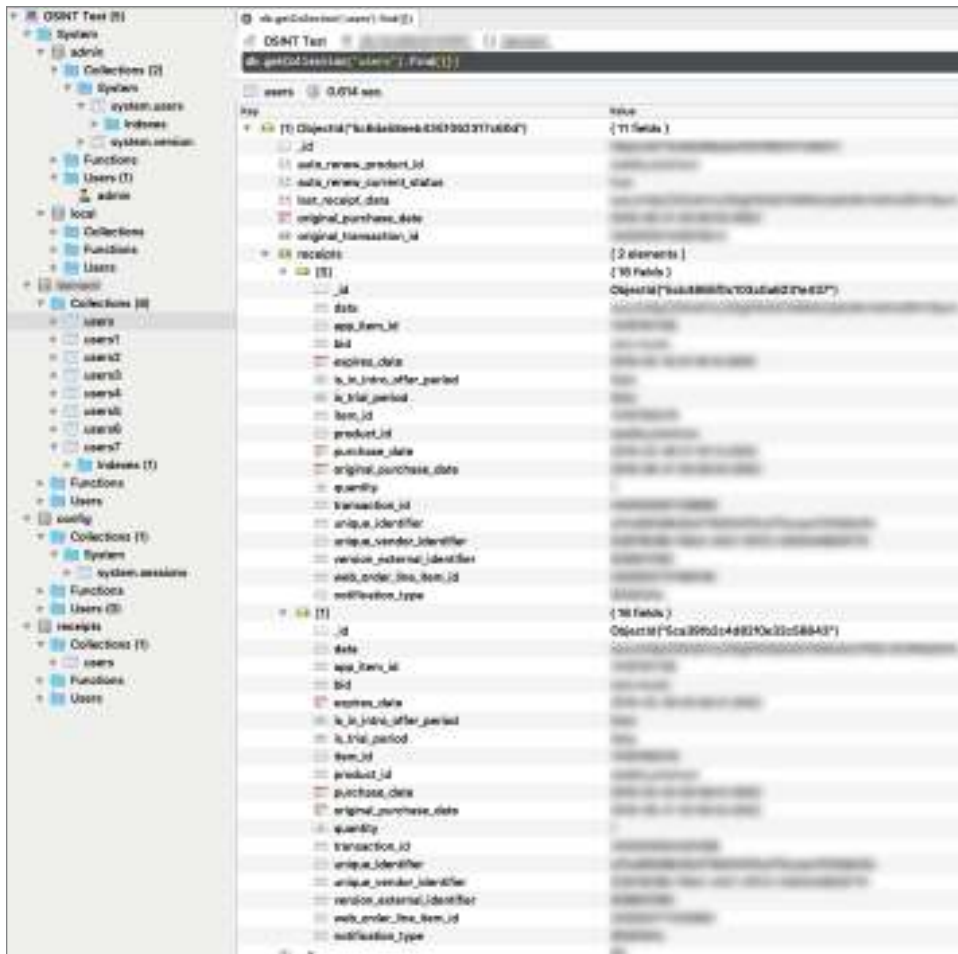


Figure 14.6



Figure 14.7

A MongoDB instance can have multiple databases. Once you have connected, you can show the different databases in the MongoDB instance using the `show dbs` command:

```
> show dbs
admin      0.000GB
bernard    2.507GB
config     0.000GB
local      0.000GB
receipts   0.001GB
```

Next, we can select a specific database to use by typing `use` followed by the database name:

```
> use bernard
switched to db bernard
```

From here, we can list the different collections within the selected database by typing `show collections`:

```
> show collections
users
users1
users2
users3
users4
users5
users6
users7
```

Selecting a collection is the same as selecting a database—using the `use` command followed by the collection name:

```
use users
```

To see all of the data in a MongoDB collection, you can use the `find()` command, or the `find().pretty()` command to view a prettified version of the JSON document:

```
db.collection_name.find().pretty()
```

This type of searching can take a lot of time since it will expand the contents of the entire database collection, which can be enormous. At this point it's usu-

ally better to dump the entire database or search through the data using a GUI tool like RobotMongo. For dumping, I developed a tool called NoScrape, which will be covered later in this chapter.

Elasticsearch

Elasticsearch is arguably the world's most popular open-source, enterprise-grade search engine. It is incredibly fast and just as simple. At its core, Elasticsearch stores data in JSON documents. Based on the free and open-source Apache Lucene information retrieval software library, and built with multi-tenancy in mind, Elasticsearch offers a distributed full-text search engine that can be accessed via the web.

Elasticsearch uses JSON and the Java API to make its features available to those who want to integrate with the solution. It is an ideal NoSQL datastore, as it supports real-time `Get` requests.

Elasticsearch distributes index operations among shards that in turn can have their own replicas. The distributed architecture makes Elasticsearch scalable by design, allowing it to provide near real-time search capabilities.

What I particularly love about Elasticsearch is how easy it is to query the data. Everything can be performed using simple `curl` commands.

Querying Elasticsearch

To query Elasticsearch, simply type `curl ip:port`. In the following example, I will query a local copy of Elasticsearch:

```
> curl localhost:9200

{
  "name" : "node1",
  "cluster_name" : "myES",
  "cluster_uuid" : "UVc8iPj4TlqVdf-IacHcOw",
  "version" : {
    "number" : "6.6.1",
    "build_flavor" : "default",
    "build_type" : "rpm",
    "build_hash" : "1fd8f69",
    "build_date" : "2019-02-13T17:10:04.160291Z",
    "build_snapshot" : false,
    "lucene_version" : "7.6.0",
    "minimum_wire_compatibility_version" : "5.6.0",
    "minimum_index_compatibility_version" : "5.0.0"
  },
  "tagline" : "You Know, for Search"
}
```

The results will show the general information for the version of Elasticsearch server running on the IP you queried. Now that we know the server is active, let's query it again to show the list of available indices (databases):

```
curl -X GET "localhost:9200/_cat/indices?v"
health status index. uuid          pri rep docs.count docs.deleted store.size
green open   n203   i98ZmZQgSA    4   1   711314316      23      162.5gb
yellow open   n204   CVBXsWyIT     5   1   385781741      0       68gb
green open   n205   HHrwpFHiQ7    4   1   211146503      0      264.8gb
```

In this list, we can see three indices: n203, n204, and n205, many containing several hundred million records (each stored in JSON documents).

Since the index names are often obscured, the next step is to investigate the mappings within each index. The mappings are similar to table headings. They will provide the structure of the index and give you a fairly good idea of what you will find inside. The output of this command will be pure JSON, so it may help to paste the data into a JSON linter to help format it for easier reading (e.g., www.jsonlint.com):

```
curl -X GET "localhost:9200/my-index/_mapping"
{"my-index":{"mappings":{"breach":{"dynamic":"false", "_all":{"enabled":false}, "properties":{"address":{"type":"text"}, "dob":{"type":"keyword", "ignore_above":256, "normalizer":"lowercase_normalizer"}, "email":{"type":"keyword", "ignore_above":256, "normalizer":"lowercase_normalizer"}, "hash":{"type":"keyword", "ignore_above":256}, "i":{"type":"keyword", "ignore_above":100}, "mobile":{"type":"text", "analyzer":"phone_number"}, "name":{"type":"text", "fields":{"keyword":{"type":"keyword", "ignore_above":256, "normalizer":"lowercase_normalizer"}}}, "password":{"type":"keyword", "ignore_above":256}}}}}}
```

The output is not easy to read, so let's copy and paste this data into jsonlint.com (Figure 14.8).



Figure 14.8

Once we do that and click Validate JSON, we get the following output:

```
{
  "my-index": {
    "mappings": {
      "breach": {
        "dynamic": "false",
        "properties": {
          "address": {
            "type": "text"
          },
          "dob": {
            "type": "keyword",
            "ignore_above": 256,
          },
          "email": {
            "type": "keyword",
            "ignore_above": 256,
          },
          "hash": {
            "type": "keyword",
            "ignore_above": 256
          },
          "ip": {
            "type": "keyword",
            "ignore_above": 100
          },
          "mobile": {
            "type": "text",
          },
          "name": {
            "type": "text",
          },
          "password": {
            "type": "keyword",
            "ignore_above": 256
          },
          "username": {
            "type": "keyword",
            "ignore_above": 256,
            "normalizer": "lowercase_normalizer"
          }
        }
      }
    }
  }
}
```

Looking at the index mappings, we can see field names like `username`, `password`, `name`, `address`, and all sorts of other useful information.

This is the *exact* process I used to discover and validate the information on Exactis, Verifications.io, and Apollo.io. In each case, authentication was not required; it took just a few simple `curl` commands.

NOTE The lack of authentication needed with Elasticsearch (and most other NoSQL databases) has given way to a huge surge in database ransomware. If you can access an Elasticsearch database, then you have full CRUD capabilities. Cyber criminals will look for open Elasticsearch databases, download all of the data, delete it from the server, then replace it with a handy ransom note. I come across these quite a bit. Yet another reason why you should always remember to put your NoSQL databases behind some sort of firewall or access control list.

Dumping Elasticsearch Data

If you find data that looks useful, you will most likely want to download a copy for examination. Doing that requires `Elasticdump`, which also requires `NodeJS` and `NPM` to be installed.

After installing `Elasticdump`, running the command is just as simple as querying the database. The only difference is that you need to specify an input and output file, and a per-query transfer limit. The maximum number of documents you can export per request is 10,000. I would suggest starting with that number and if you find that the server is sluggish in serving your requests, try dropping the value until you find something that works for you. The following command will begin dumping an Elasticsearch database from server 0.0.0.0:

```
> elasticdump --input=http://0.0.0.0:9200/index-name \\  
--output=index-name.json --noRefresh --limit=10000  
  
starting dump  
got 10000 objects from source elasticsearch (offset: 0)  
sent 10000 objects to destination file, wrote 10000  
got 10000 objects from source elasticsearch (offset: 10000)  
sent 10000 objects to destination file, wrote 10000  
dump complete
```

Your output will look similar to this code and when complete, you will have a JSON file with the contents of the requested index.

NoScape

NoScape is an open-source tool designed to ease the process of locating and scraping data sets available in open storage containers and publicly accessible databases. Initially developed by William Martin and me, NoScape is now

available as a free download from the Night Lion Security Gitlab page: <https://gitlab.com/nightlionsec/noscraper>.

At the time of writing this section, NoScrape supports easily accessing and downloading data from MongoDB, AWS buckets, Elasticsearch, and CassandraDB.

We designed the tool because the process of identifying useful data within MongoDB and Elasticsearch was extremely cumbersome when dealing with tens of thousands of active hosts. We also added the ability to search through AWS S3 buckets on an extremely large scale—all you need is a wordlist.

NoScrape has a lot of parameters that can be specified depending on the function you are performing.

Arguments include:

```
-d Database Type           : Type of database to scrape
                           : ["mongo", "s3", "es", "cassandra"]
-o OutputFile              : File to output the results
-v                          : Enable verbose output
--examples                 : Print usage examples then exit
```

Mongo And Cassandra Options

```
-tf, --targetFile          : CSV formatted list of hosts (IP,port)
-t, --target Target        : IP or CIDR of database
-p, --port Port            : Port of database(s)
-s, --scrape               : Type of scrape to run: basic, full
-f, --filter FilterFile    : A file containing a list of keywords
to match
-u, --username Username    : Username for standard authentication
-p, --password Password    : Password for standard authentication
-a, --authDb AuthDB        : Database within Mongo to auth against
Elasticsearch Options
```

```
-tf, --targetFile          : CSV formatted list of hosts (IP,port)
-t, --targets Targets      : IP or CIDR of database(s)
-p, --port Port            : Port of database(s)
-s, --scrape               : Type of scrape: scan, search, dump,
                           matchdump
-f, --filter FilterFile    : A file containing a list of
                           keywords to match
-l, --limit Limit          : Number of ES records to dump per
                           request (default / maximum: 10000)
```

AWS Brute S3 Options

```
--access AccessKey      : AWS Access Key
--secret SecretKey      : AWS Secret Key
--hitlist DictionaryFile : File containing a list of words to try
```

MongoDB

Starting with MongoDB, running NoScrape to grab data from a MongoDB server is very similar to the MongoDB command-line tools. The difference is having many of the scraping queries automated for you, so you do not have to dump each collection and database individually.

When scraping MongoDB, NoScrape allows two basic scrape types: basic and full. A *basic scrape* will list all of the databases and tables/collections in a tree-list format. A *full scrape* will list all of the databases, all of the tables/collections within each database, and all of the fields within each collection.

To start scanning a list of MongoDB IPs, specify the database type to `mongo` using the `-d` parameter, and a CSV-formatted list of IPs and ports using the `-tf` (targetfile) parameter. Be sure to use the `-o` parameter and also specify your output file. This is the same as also using the standard Linux parameter to send all output to a file: `> outputfile`.

The following command will start a basic dump of the IPs located in a `mongolist.txt` file with a basic scan type:

```
./noScrape.py -d mongo -s basic -tf mongolist.txt
```

I did not specify an output file so I can see the output in real time. If you are scanning a lot of databases, this will fill up your screen buffer very quickly.

Looking through dumped MongoDB collections and tables was already covered earlier in this chapter, so we can skip over that.

Ransomed MongoDB Server

It's worth mentioning before we move on to Elasticsearch that if you are using NoScrape to randomly scrape a number of different MongoDB IP addresses, there is a good chance your output file will also contain a number of error messages. Here is what some of the more common error messages will look like:

```
Attached to 0.0.0.0:27017
{DB} hacked_by_unistellar
Collection} restore
```

As mentioned earlier, you may come across NoSQL databases that have been removed and are now being ransomed. This is what it will look like in a MongoDB.

No Authorization

```
An unexpected error occurred while enumerating the databases on
0.0.0.0:27017 - not authorized on admin to execute
command { listDatabases: 1, nameOnly: true }
```

You will see this message if the MongoDB server has authentication set up. More and more MongoDB instances have authentication now, which means even if the database is detected by Shodan, you will not be able to access it without proper credentials.

Bad IP/Timed Out

```
Error connecting to 0.0.0.0:27017 - timed out
```

Pretty self-explanatory. The IP is no longer active or a firewall has been set up to prevent access.

Elasticsearch

NoScrape was initially developed as a way to quickly search and dump a large number of Elasticsearch databases. Everything that NoScrape does can just as easily be done manually using `curl` commands (as discussed earlier in this chapter). The difference, again, is that NoScrape puts everything in one nice and neat package for you.

Similar to the MongoDB options, I personally specify a target CSV file that contains a list of IPs and their ports. When running NoScrape, make sure to set the database type to `es` using the `-d` parameter.

The Elasticsearch scraper within NoScrape has four different scrape types that can be specified using the `-s` parameter: `scan`, `search`, `dump`, and `matchdump`. Let's look at them.

Scan

The `scan` parameter will list the basic output of each Elasticsearch server and output the contents of the indices to the screen (or output file). This is the same as querying the server using the `/_cat/indices?v` command discussed earlier:

```
[ root@osint ] > python3 noscrape.py -d es -tf es-list.txt -s scan

----Results: 0.0.0.0:9200
health status index uuid pri rep docs.count docs.deleted store.size
yellow open   nginx-dos-router-2019.04.29 1oE... 5 1 3773305 0 1.5gb
yellow open   app-dos-web-2019.03.31      1Nq... 5 1 2702331 0 892.12
yellow open   nginx-dos-web-2019.05.01    3B0... 5 1 9851 0 2.9mb
yellow open   nginx-dos-web-2019.04.04    EaA... 5 1 10023 0 3mb
```

That is the typical ES output from one server. Notice at the beginning of the file there is `---Results: IP`. This will hopefully make it easier to search and clean up this list later on.

A process you might want to use is to search for certain keywords and remove everything else. You can achieve this using the `--filter` parameter. I don't recommend using that option at this particular stage of the search because you will miss a lot of important results if you are only looking at an index's name.

Search

The `search` parameter will perform a deeper dive into the Elasticsearch database output by looking at the mappings table for each of the indices within the database. This is the same as looking at a table's headers. Using this option is useful because more often than not, the actual index name will be obscured or not something that would ever pique interest. However, the mappings file will show you exactly what type of data is located within each table.

The `dump` parameter will use `Elasticdump` to dump every index within the IP address. Use this with caution as it will download everything. You do not need to specify an output filename since `Elasticdump` will use the name of the index as the JSON filename.

There is a very good chance that the output for this will be massive, so be prepared to deal with an extremely large file. An output will look like this:

```
[ root@osint ] > python3 noscrape.py -d es -tf es-list.txt -s search
```

```
{ "type": "keyword", "timezone": { "type": "keyword" } }, "host": { "type":
"keyword", "hostname": { "type": "keyword", "jsessionId": { "type":
"keyword", "level": { "type": "keyword", "logger_name": { "type":
"keyword", "message": { "type": "text", "messageId": { "type": "keyword",
"requestUri": { "type": "keyword", "requestedSessionId": { "type":
"keyword", "requestedSessionIdFromCookie": { "type": "keyword",
"requestedSessionIdValid": { "type": "keyword", "service": { "type":
"keyword", "sessionId": { "type": "keyword", "stack_trace": { "type":
"text", "subscriptionId": { "type": "long", "system": { "type": "keyword",
"tags": { "type": "keyword", "thread_name": { "type": "keyword", "type":
{ "type": "keyword", "user": { "type": "keyword", "userId": { "type":
"keyword", "userIp": { "type": "keyword", "useragent": { "dynamic":
"true", "properties": { "build": { "type": "keyword", "device": { "type":
"keyword", "major": { "type": "keyword", "minor": { "type": "keyword",
"name": { "type": "keyword", "os": { "type": "keyword", "os_major": { "type":
"keyword", "os_minor": { "type": "keyword", "os_name": { "type":
"keyword", "patch": { "type": "keyword" } }, "x-forwarded-for":
{ "type": "text", "x-forwarded-host": { "type": "keyword",
"x-forwarded-port": { "type": "keyword", "x-forwarded-proto":
{ "type": "keyword", "x-forwarded-server": { "type": "keyword",
```

```

"x-prerender-token":{"type":"keyword"},"x-real-ip":{"type":
"keyword"},"x-request-id":{"type":"keyword"}}}, "ERROR":{"_all":
{"enabled":false,"dynamic_templates":[{"message_field":{"match":
"message","match_mapping_type":"string","mapping":{"index":"analyzed",
"type":"string"}}}, {"string_fields":{"match":"*","match_mapping_type":
"string","mapping":{"index":"not_analyzed","type":"string"}}}],
"properties":{"@timestamp":{"type":"date"},"_ga_cid":{"type":"keyword"},
"_mxpnl_cd":{"type":"keyword"},"_mxpnl_cm":{"type":"keyword"},
"_mxpnl_cw":{"type":"keyword"},"accept":{"type":"keyword"},
"accept-encoding":{"type":"keyword"},"accept-language":{"type":
"keyword"},"agent":{"type":"keyword"},"application_name":{"type":
"keyword"},"application_version":{"type":"keyword"},"cf-connecting-ip":
{"type":"keyword"},"cf-ipcountry":{"type":"keyword"},"cf-ray":{"type":
"keyword"},"cf-visitor":{"type":"keyword"},"clientip":{"type":
"keyword"},"connection":{"type":"keyword"},"dos_ga_clientid":{"type":
"keyword"},"dos_mixpanel_clientid":{"type":"keyword"},"environment":
{"type":"keyword"},"geoip":{"dynamic":"true","properties":{"area_code":
{"type":"long"},"city_name":{"type":"text"},"continent_code":{"type":
"keyword"},"coordinates":{"type":"double"},"country_code2":{"type":
"keyword"},"country_code3":{"type":"keyword"},"country_name":{"type":
"keyword"},"dma_code":{"type":"long"},"ip":{"type":"keyword"},
"latitude":{"type":"float"},"location":{"type":"geo_point"},
"longitude":{"type":"float"},"postal_code":{"type":"keyword"},
"real_region_name":{"type":"keyword"},"region_code":{"type":"keyword"},
"region_name":{"type":"keyword"},"timezone":{"type":"keyword"}}},
"host":{"type":"keyword"},"hostname":{"type":"keyword"},"jsessionId":
{"type":"keyword"},"level":{"type":"keyword"},"logger_name":{"type":
"keyword"},"message":{"type":"text"},"planId":{"type":"keyword"},
"requestUri":{"type":"keyword"},"requestedSessionId":{"type":"keyword"},
"requestedSessionIdFromCookie":{"type":"keyword"},
"requestedSessionIdValid":{"type":"keyword"},"service":{"type":
"keyword"},"sessionId":{"type":"keyword"},"stack_trace":{"type":
"text"},"subscription":{"type":"keyword"},"system":{"type":"keyword"},
"tags":{"type":"keyword"},"thread_name":{"type":"keyword"},"type":
{"type":"keyword"},"user":{"type":"keyword"},"userId":{"type":
"keyword"},"useragent":{"dynamic":"true","properties":{"build":
{"type":"keyword"},"device":{"type":"keyword"},"major":{"type":
"keyword"},"minor":{"type":"keyword"},"name":{"type":"keyword"},"os":
{"type":"keyword"},"os_major":{"type":"keyword"},"os_minor":{"type":
"keyword"},"os_name":{"type":"keyword"},"patch":{"type":"keyword"}}},
"x-forwarded-for":{"type":"text"},"x-forwarded-host":{"type":"keyword"},
"x-forwarded-port":{"type":"keyword"},"x-forwarded-proto":{"type":
"keyword"},"x-forwarded-server":{"type":"keyword"},"x-prerender-token":
{"type":"keyword"},"x-real-ip":{"type":"keyword"},
"x-request-id":{"type":"keyword"}}}, "...

```

This is a huge mess, but if you look closer at the results, you will notice something really interesting. The headers include keywords like `ip`, `latitude`, `geoip`, `areacode`, and `userid`.

NOTE I am being completely honest when I say I just pulled this IP at random and had no idea what was there. This looks really interesting, though, and worth exploring. See how fun this is?

Because there are so many results, this would be a good stage to use the `--filter` parameter. Using the filter allows you to specify a file of keywords to match the results. An output will only be displayed if one or more of your keywords match the search results.

In this case, the preceding output will contain matches for several keywords, so the output would be exactly the same. Once you have the data downloaded, it is typically a manual process to look through the results, or you can skip this step altogether and go right to dumping all of the data.

Dump

The `dump` command will just dump all the data on every IP you select. If you just want everything and don't care about what you are downloading, this option is for you.

The maximum (and default) number of records that can be dumped with each Elasticdump request is 10,000. If you encounter slow servers, you can try lowering this using the `-l` (limit) parameter.

The `dump` command will automatically create a folder for each IP address being dumped. All downloaded data will be stored in that folder. The following command will run NoScrape against any IP address located in the `es-test.txt` file, and automatically download all of the databases regardless of content:

```
./noScrape.py -d es -tf es-test.txt -s dump
```

If the command is successful in connecting to a target IP, the output will look like this:

```
GET http://0.0.0.0:9200/* [status:200 request:0.116s]
Dumping to file for ip 0.0.0.0:9200 and the index nginx... 2019.01.29

http://0.0.0.0:9200/* [status:200 request:0.116s]
Dumping to file for ip 0.0.0.0:9200 and the index nginx...2019.02.29

GET http://0.0.0.0:9200/* [status:200 request:0.116s]
Dumping to file for ip 0.0.0.0:9200 and the index nginx...2019.03.29
```

Your output will be located in the `(0.0.0.0:9200)` folder, and will be a list of JSON files downloaded from each server.

MatchDump

The `matchdump` parameter is a combination of all the previous scan and scrape methods. Rather than taking a completely broad approach of just downloading

everything and having to manually go through it all later, `matchdump` tries to simplify this process by only dumping databases that match specific criteria.

The process performs multiple searches against the Elasticsearch database, starting with a combination of the “search” and “filter” functions. During the first step in the process, NoScape will grab all of the mappings data for the Elasticsearch indices and check them against your filter keyword list. Using this option requires you to specify the `--filter` parameter.

Once keywords are matched, the next step is for NoScape to use the `dump` parameter and download all of the matched databases.

Running the full command looks something like this:

```
./noscape.py -d es -tf list.txt -s matchdump --filter keywords.txt
```

When we execute this command, the output should look exactly like the results of the previous `dump` example. In this case, the random ES server that I picked happened to hit many of the keywords I normally use, including username, IP, geoIP, address, etc.

Once those keywords are matched, the `matchdump` option tells NoScape to also dump the entire database. The files will be stored in a folder according to the server’s IP address.

When building a search list to look for interesting data fields, I recommend the following keywords as a good starting point:

- Password
- Username
- Email (this will also match `email_address` and any variation)
- GeoIP
- IP
- FacebookID
- LinkedIn
- Instagram
- Hash
- Salt
- Telephone, Mobile, Cell

Cassandra

Apache’s Cassandra is another popular NoSQL database type that is very similar to MongoDB. Because of the similarities in database types, the Cassandra options of NoScape are exactly the same as MongoDB. The only difference in application use is that running a scan or scrape against a Cassandra DB requires you to change the `-d` parameter to `Cassandra`.

The following command will run a basic scan against a list of CassandraDB IPs:

```
./noScrape.py -d cassandra -s basic -tf cassandra-list.txt
```

A successful run will look very similar to MongoDB:

```
[2019-05-12 08:02:15] New Cassandra host <Host: 0.0.0.0 datacenter1>
discovered
[2019-05-12 08:02:19] {Keyspace} weather_geohash_ks
[2019-05-12 08:02:19]     {Table} GRIB2_Data
[2019-05-12 08:02:19]     {Table} GRIB2_PIC
[2019-05-12 08:02:19]     {Table} GRIB2_Status
[2019-05-12 08:02:19]     {Table} GRIB2_Status_HTSWG
[2019-05-12 08:02:20] Failed to create connection pool for new host
172.17.137.165:
OSError: [Errno None] Tried connecting to [('0.0.0.0', 9042)].
Last error: timed out
[2019-05-12 08:02:20] Using datacenter 'DC1' for DCAwareRoundRobinPolicy
(via host '0.0.0.0'); if incorrect, please specify a local_dc to the
constructor, or limit contact points to local cluster nodes
[2019-05-12 08:02:21] {Keyspace} twitter
[2019-05-12 08:02:21]     {Table} user
[2019-05-12 08:02:21] {Keyspace} elastic_admin
[2019-05-12 08:02:21]     {Table} metadata
[2019-05-12 08:02:21] {Keyspace} dbtest2
[2019-05-12 08:02:21]     {Table} alumno
[2019-05-12 08:02:21]     {Table} alumnos
[2019-05-12 08:02:21]     {Table} ambulancia
[2019-05-12 08:02:21]     {Table} area
[2019-05-12 08:02:21]     {Table} aseguradora
[2019-05-12 08:02:21]     {Table} asistencias
[2019-05-12 08:02:21]     {Table} autentificacion
[2019-05-12 08:02:21]     {Table} auto
[2019-05-12 08:02:21]     {Table} cajas
[2019-05-12 08:02:21]     {Table} cama
[2019-05-12 08:02:21]     {Table} categoria
[2019-05-12 08:02:21]     {Table} centrocosto
[2019-05-12 08:02:21]     {Table} citas
[2019-05-12 08:02:21]     {Table} cliente
[2019-05-12 08:02:21]     {Table} consultorio
[2019-05-12 08:02:21]     {Table} convenio
[2019-05-12 08:02:21]     {Table} criterioevaluacion
[2019-05-12 08:02:21]     {Table} curso
[2019-05-12 08:02:21]     {Table} cursoprogramado
[2019-05-12 08:02:21]     {Table} diagnostico
[2019-05-12 08:02:21]     {Table} dietas
[2019-05-12 08:02:21]     {Table} docente
```

From there, you can choose to run a more in-depth scrape type to see the collection data, or just dump everything using the `-d` parameter.

Amazon S3

NoScrape also supports scanning for Amazon S3 buckets. The main difference between NoScrape and CloudStorageFinder (discussed earlier in this chapter) is the ability to specify authentication for AWS.

Using your own S3 authentication can prove to be an easy way to find buckets with misconfigured authentication settings.

Using Your Own S3 Credentials

Several months ago, a particular reporter wrote a story about how I contacted him regarding the exposure of All American Entertainment's S3 bucket. AAE happens to be this reporter's main speakers bureau, so the leak included his speaking alongside many other high-profile celebrities and public figures, including Dwayne "The Rock" Johnson, Gwyneth Paltrow, Hillary Clinton, Colin Powell, and many more.

The reporter was perturbed enough by this discovery to include the details in his story. However, following the publication of the story I've come to realize that the reporter was more perturbed with me than the actual discovery.

Regardless, AAE's "misconfiguration" was somewhat complex (at the very least, confusing) and would not have been easily caught during setup by someone who did not have an advanced knowledge of Amazon's overly confusing configuration settings.

What I am saying is that certain S3 buckets that require authentication can be accessed using *any* authentication—this includes your own S3 access keys.

In other words, *I was able to discover the AAE leak by accessing AAE's private bucket using my own credentials.*

This is why NoScrape was developed with the ability to perform wide-scale S3 bucket discovery using any credentials.

The following command will perform a scan of S3 buckets, using a specific Amazon access key and secret key:

```
[ root@osint ] > python3 noScrape.py -d s3 --access AccessKey \\  
--secret SecretKey -tf wordlist.txt
```

If the bucket exists but the AWS key was invalid (or not authorized), you will see the following error:

```
[S3_NoAccess] 's3://acciones' exists, but we do not have access  
[S3_NoAccess] 's3://actividad' exists, but we do not have access  
[S3_NoAccess] 's3://vinnytroia' exists, but we do not have access
```

On the other hand, if the key was successful (or if you are just scanning for open buckets with no authentication), you will see a message similar to the following:

```
Identified access to 's3://vinnytroia-test' -  
Listing all objects/files...  
- OSINT-book-test.txt  
S3 Module Completed
```

Looking at the results, we can see that one of the buckets on our list, `vinnytroia-test`, is publicly accessible. This is a bucket I created for this example. The one file listed is `OSINT-book-test.txt`. If this were a real S3 bucket, there would probably be more files listed.

NoScrape is not set up to download the bucket files (yet), but that feature will hopefully be complete by the time this book is published. In the meantime, sending all output to a file will allow you to quickly identify and target buckets with active listings.

Summary

This chapter covered a lot of information on how to find publicly accessible data sets. It provided information on searching through public cloud storage containers like Amazon S3 and Digital Ocean, an introduction to NoSQL database types like MongoDB and Elasticsearch, and commands to manually query and download files from those databases. The NoScrape tool provides automation of complex and tedious tasks like searching, scraping, and dumping data from several different types of NoSQL databases, including MongoDB, Elasticsearch, and CassandraDB.

Now that we have identified ways to find data on publicly accessible servers, the next section of this book will focus on hunting people and the different tools we can use to find information on human targets!

People Hunting

In This Part

Chapter 15: Researching People, Images, and Locations

Chapter 16: Searching Social Media

Chapter 17: Profile Tracking and Password Reset Clues

Chapter 18: Passwords, Dumps, and Data Viper

Chapter 19: Interacting with Threat Actors

Chapter 20: Cutting through the Disinformation of a 10-Million-Dollar Hack

This section focuses on the art of investigating people. The next few chapters will cover a wide array of technologies and techniques that range from simple social media research to leveraging information contained in breached databases. Throughout this section, I will also cover building a “tracking matrix,” which will help keep your information organized and, if done properly, will also help you organize your research in a way that will help you visually spot clues faster.

What better way to start this section than with a tip from the “Human Hacker”?

EXPERT TIP: CHRIS HADNAGY

When conducting an investigation or a penetration test, there really isn't a one-size-fits-all model for investigation tools and techniques that will work every time. It really depends on what the target goal is. Is it a spear phish or general phishing? Are we OSINTing a whole company or just a person? Or are we trying to get a shell versus just a click?

There are just so many avenues. If we're trying to get a shell, let's say, it may be a multi-staged attack for us when it comes to OSINT. We take the time to research and find something where it makes sense for them to enter credentials. We look for something in their profiles and their social media that allows us to pretext as somebody that would legitimately be asking for credentials. Then, once we get the credentials, we might follow up from whatever the pretext is, with a document that's loaded with some kind of malware that will give us a shell.

Whereas if we're just going for a click, sometimes, companies just want us to see if they're susceptible to clicking. In that case, we'll look for the biggest emotional triggers we can find, such as charities, hobbies, kids' events, and things like that.

If it's corporate OSINT, then it's more than just trying to figure out how they use social media, because everybody does. With corporate OSINT, we're looking through Flickr and Instagram and their corporate Facebook page to see if they do things like the company Christmas party or the year-end picnic or whatever it is, and all the folks have their badges on. Or, for example, if they had a big announcement or a new product release, and the company hosted a big party where people are taking pictures and they are still wearing their badges.

Then, we might look for employees that maybe just got hired, and they're excited and it's like, "Look, Mom, my new desk," and it has a picture of their laptop turned on with their desktop up and their badge and their phone and everything that they have at their desk.

They don't realize how much they are sharing in those photos.

Researching People, Images, and Locations

We finally come to my personal favorite part of an investigation: looking into people. I don't know what it is, but there is something incredibly satisfying about finally learning who your targets are, behind all the layers of aliases and alter egos. Once you are able to find that magical strand of evidence to unravel, it's like a new world opens for you. Time begins to slow down and suddenly everything becomes clear. It's like you are in bullet-time from *The Matrix*. It's incredible.

EXPERT TIP: JOHN STRAND

Anytime you're trying to target individuals in an organization, you have to have a process for identifying the most probable targets to interact with that will click on a link or trust something you send them.

What I train all of my testers and everybody that goes through my SANS course is to always look for bright, shiny objects. So, imagine that you have an organization that has 4,000 individuals; the individuals that are the most interesting to me are the individuals that are the most heavily engaged in social media.

They're the individuals that have a very heavy presence on Twitter, they're the people that constantly take pictures of their food and they post it online. They're the people that have opinions and insist on those opinions being present someplace out on the web. When you're building up a pretext to socially engineer an individual, it

is really interesting because you can use all of this information to very quickly get a person to trust you.

You usually only need about three points of contact that are associated with trust for them to start trusting you completely, because human beings usually can only handle about 150 interpersonal relationships that you can actually care about at one time. Beyond that, it just doesn't matter.

For example, if someone hears about a ferry that goes down in the Mediterranean and 400 people die a horrible death, you're like, "Oh, that sucks." But it really doesn't matter to them and they can easily move on with their lives. Whereas if your dog gets hit by a car, you're inconsolable for weeks. Your brain can only have the capacity to handle so many social interactions at one time.

For that, we actually have shortcuts that we can employ to quickly elevate ourselves during an engagement. One of the easiest things to do if I want to get you to agree with me is that I will identify your religious or political preferences and then I can reflect that back to you.

I'm reflecting *you* back to you and in a way, the target is going to fill in the rest of that straw man. Once you get about three points of reflection, then they start thinking, "Well, this person thinks just like me and I'm a good person, so this person must be a good person. I'm trustworthy and this person must be trustworthy, too." It becomes very easy to get them to click on links, to go to web pages, or to interact with you at a much more trusting level than just saying, "Hey, here's 50 percent off shoes at some department store."

I mean, it just takes a little bit of effort, but it makes you target-rich. So you're trying to identify the users that are the most likely to fall into those traps, and the ones that are the most active on social media are the ones that are generally the most enticing targets because they have this deep need, and you're going to feed that need. That need is very simple: they just want to be seen and recognized. If you can see and recognize them and reflect back their virtues or virtue-signaling, like a reflective virtue-signaling to these people, then they will trust *anything* you say or do.

PIPL

I *love* PIPL (www.pipl.com). PIPL (pronounced "people") is arguably the world's largest people-related search engine (according to its own website). I don't know whether or not that is actually true, but the service is amazing, and it is usually my first stop when I am trying to find additional information about a person.

PIPL allows you to search by an email address, social username, and/or phone number. It also contains a wealth of information about people's associated social media information, usernames, and alternate email addresses.

EXPERT TIP: CAT MURDOCK

I feel like I grew up with the Internet. I went to college and Facebook became a thing while I was there, so I've kind of gotten to grow up alongside some of these platforms. Social media, to me, often feels like low-hanging fruit, so I will generally start my investigation there . . . because, why not?

I'm a very visual person. I think that there is a lot of power in photos, and I'm sure as you recognize, the information presented may not be exactly what the person sharing that information thinks it is. For example, you may suddenly see an outlet in the background of a photo, and you know the nation that the person is in.

That may not have been something that they thought about . . . So I will start with the low-hanging fruit of scouting social media sites. I like to do an overall audit of any potential email addresses I can find, any usernames I can find, and any handles that they may be operating with online. And then, depending on how robust all of those data points are, I'll also try to consult what Nick Furneaux referred to as their "digital shadow."

But let's use my mother as an example. She has a unique last name, and so, if I were going to try to OSINT her (or myself), I would first look to see what kind of results PIPL has, which is, quite frankly, a toy I love.

I don't know if PIPL scraped data from Facebook before it shut down part of its search parameters, but if you search a phone number in PIPL, and it is associated with a Facebook profile, it will show you that Facebook profile, even though Facebook will no longer do that for you. And, if you search PIPL for a phone number, there is a good chance you will find basic user profile information even if that profile has been previously deleted from social media.

I also love using PIPL with Maltego—give it a couple of searches and you will get back a whole list of phone numbers and accounts.

But there are so many false positives; you really do still have to do your own homework, and make sure that those profiles are real or even valid. So I absolutely love Maltego as an initial tool to just do some initial discovery.

I think that the desire from so many people is to have it be the end-all-be-all answer, and it's just not—nothing ever is. You always have to put in the effort one way or another.

Searching for People

PIPL is incredibly easy to use, and its API is not very expensive for the value it provides. For those using Maltego, PIPL's API is built into several of the key social media research plugins. For those not using Maltego, the website is extremely straightforward. Figure 15.1 shows the basic PIPL search page.



Figure 15.1

To kick off our search, let's use me as the target. Until now, I have never searched for myself in PIPL, so the results are kind of interesting to see.

NOTE In my former life, I started and ran a company called Curvve Recordings, an electronic dance music record label (www.curvverecordings.com). I also produced music and toured as a DJ. If you do decide to search for me, many of the results you may find are related to music.

Figure 15.2 shows the results of using “Vinny Troia” in the PIPL search.

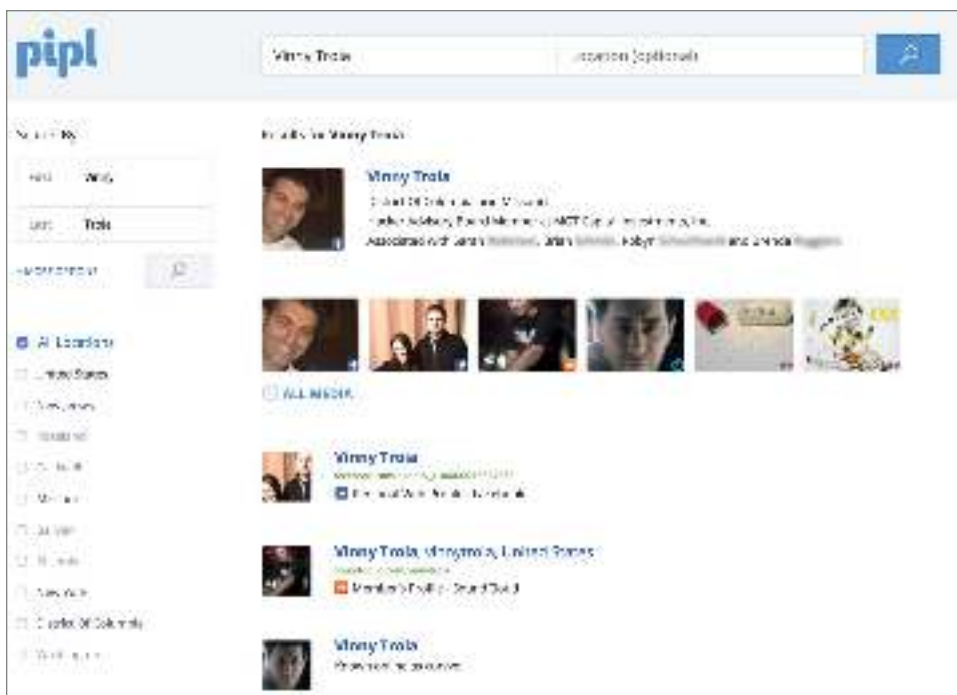


Figure 15.2

The results are interesting—not exactly accurate, but a pretty decent account of my online personas. For one, I don't live in Washington, DC. The Facebook

page that came up is not mine either, but the SoundCloud page does belong to me. The remaining results (which were cut off) are related to musical works.

I feel the “associated with” section (three lines under my name) is the most interesting data on this page. Not because it is overly accurate, but because of how random it is. The first person I know, very loosely. The second person is someone whose number is in my cell phone, but we haven’t talked in years. The third person I have never heard of, and the fourth person is someone I went to high school with, but have not spoken to in about that long.

This is probably the most random grouping of “associates” I have ever seen, because while I know most of these people, I can’t say that I am really associated with any of them.

Though these results may not seem to provide the best example of how much data is stored in the PIPL database, it should give you an idea of the *type* of results PIPL can provide. This is also a great example to point out that data being presented (by any service) may not be entirely accurate, which is why everything should be verified.

I would also never discount anything. You would be surprised how common it is to find a threat actor’s SoundCloud page using a profile icon (or username) that shares similarities to an account or profile on a different website.

To illustrate a similar point, let’s move back to search for our threat actors. I remember one day I decided to try my luck and search for “TheDarkOverlord” in PIPL. Figure 15.3 shows the results.



Figure 15.3

Much to my surprise, there was a match! Poor Craig. I wonder how many calls he has received as a result of this information?

Full disclosure—As far as I know, TheDarkOverlord is *not* Craig. But I find it really interesting that Craig would even be associated with the name. Clicking the link, Figure 15.4 shows us why.

Evidently, Craig at some point had a personal eBay and Netlog page under the name “TheDarkOverlord.” Neither page exists anymore, but PIPL has them cached and stored. Even though this information did not actually help us, when you think about what just happened it is actually really powerful.



Figure 15.4

PIPL at one point cached these pages and associated this person to TheDark-Overlord based on his username. Even though those pages no longer exist, we can see they existed *at one point*.

Access to historical data can be (and often is) the most crucial part of an investigation—so having access to this kind of information, for literal pennies per search result, is unbelievable. While this turned out to not be relevant, there is a good chance PIPL will have other matches in the future that are more useful.

NOTE PIPL recently changed its user interface, adding a slew of new options and features. The new changes were added after most of this book's contents were written, so I could only sneak in this extra note.

That being said, the new advanced search capabilities are fairly significant, now allowing you to filter and pivot your search results to include phone numbers, usernames, addresses, associate names, education, jobs, and age.

Public Records and Background Checks

Hundreds of websites allow you to run background checks on people. If you are looking for a simple run-of-the-mill background check, I recommend these two websites:

- FreeBackgroundCheck.org
- SkipEase.com (which searches BeenVerified, WhitePages, and PeopleLooker)

FreeBackgroundCheck.org is, as the name implies, a free background check. If you are looking for absolutely free, it's worth a shot.

SkipEase.com is very similar to Kayak.com (the travel website) in that it provides the ability to search multiple sites at once. The appeal of Kayak is that you can search many of the major airfare and hotel sites (like Expedia and Travelocity) at the same time, allowing you to find the best deal or rate for your trip.

Similarly, SkipEase.com allows you to run simultaneous searches against BeenVerified, WhitePages, and PeopleLookup, three of the largest (and most reputable) people finder sites on the Internet. BeenVerified is typically my first stop, but since SkipEase allows you to search all three, there is no reason not to start there.

Figure 15.5 shows the SkipEase UI with a starting search for “Vinny Troia” under BeenVerified.

The screenshot shows the SkipEase People Search interface. At the top, the logo "skipease" is in red, with "People Search" in blue below it. The interface is divided into three columns, each representing a different search engine: BeenVerified, PeopleLookup, and White Pages. Each column contains a form with input fields for "First Name", "Last Name", and "State" (or "City, State or ZIP" for White Pages). Below each set of input fields is a green "Search" button. At the bottom of each column, there is a small line of text: "We search for you on the following sites: BeenVerified, WhitePages, PeopleLookup, and others." The "BeenVerified" column has a dropdown menu for "State" with "New Jersey" selected.

Figure 15.5

Each of these three sites will require you to purchase a report or pay for access to view your results, so you should be prepared to spend money here if you want to view this level of information on a target. This is another instance where you always get what you pay for.

These tools will cover your basic people searching and background checks. Following are a few other tools worth looking at that are less common.

Ancestry.com

Ancestry.com is almost always overlooked. I don't know why. It literally advertises having over 20 *billion* records to search from—by far the largest amount of historical information on people, anywhere. That sounds to me like a pretty good place to look!

What's even better about Ancestry.com is that you can purchase different packages that allow you to search international records. While not every country's records will be as thorough and up-to-date as those of the United States or Canada, this is an avenue worth checking if you have an idea of a threat actor's name or the name of a family member.

The great thing about Ancestry.com is you can get a 14-day free trial. If you don't find you use it often enough to justify a membership, you can just sign up for a new trial when you need it again.

Threat Actors Have Dads, Too

There was an instance where using Ancestry.com worked out really well for me. I was looking into a threat actor in Canada but was not able to find much since he is a minor. I loosely knew his name, though, and wanted to find out more about people he could be connected with, so I checked Ancestry.com. From there, I was able to find a small family tree—enough to at least identify a father and sister. As it turns out, his father is a very prominent and well-known chef in Canada, who incidentally had a much less restrictive Facebook page than his son. Lucky for me, I was able to look through his pictures without sending him a friend request, which is how I was able to finally find a few pictures of the threat actor in question. Boom!

Criminal Record Searches

Each state should have a public record system that you can search. If you are looking for a criminal in the United States, there is a chance they will already have a record. If you know the person's name and want to know more about their background, this is a good place to look.

The actual court website will vary between states, so a good place to start is the National Center for State Courts, at www.ncsc.org. To find a direct link for each U.S. state, go to this website:

<https://www.ncsc.org/topics/access-and-fairness/privacy-public-access-to-court-records/state-links.aspx>

Or just do a Google search for the following: "State court websites ncsc.org." The preceding URL should be the first result.

Figure 15.6 shows the NCSC website where you can browse the different court lookup tools for each state.



Figure 15.6

Image Searching

When I was investigating the origins of The Dark Overlord, there were several images that first appeared as profile avatars. While this was seemingly insignificant at first, a closer look revealed that the images were distinct enough to be linked to other online handles. Hackers (at least while they are still skids) will reuse profile images across forums, so being able to find multiple occurrences of those images, especially over social media, can be significant.

I typically turn to three main tools when searching images: TinEye, EagleEye, and Google reverse image search. Let's look at all three, but first, a word from Cat Murdock.

EXPERT TIP: CAT MURDOCK

One thing I've encountered multiple times is women who are now professionals who had compromising photos of themselves leaked online, earlier in their lives.

I think many times they don't know the data is even out there. They don't realize that someone put it there, like a photo published to a Flickr page, or someone's blog, and suddenly there it is on Google Images.

I think it's an interesting benefit of OSINT that we've been able to identify some of that information, and even help people get it removed. It's different if somebody is exploiting or extorting a person, or is actually using the data to manipulate someone in some way . . . it becomes a much more difficult situation because of all the emotions involved; it's never an easy solution to begin with.

It's been really fulfilling to be able to help people in those situations. Unfortunately when researching someone's digital shadow, you will find the images were used in some different way, like they have been cropped in a way that maybe isn't so desirable or somehow changes the intent of the image. So when you can find those images as part of an investigation, and use that information to ping Google and say, "Hey, we own this content. Can you take it down?" that's a powerful benefit of OSINT.

A reverse image search can help, or a tool like TinEye can help. Sometimes, you definitely have to look into using the different sizes option when you run the searches, because generally, they've been cropped or altered.

A couple of times where this happened is, I have seen a photo that looked like the person, or it wound up being that a former photographer had taken photos, and then used them in a way that was not approved.

Or maybe an ex-boyfriend used a photo but cropped just a face, or maybe super-imposed his face on someone else's body to appear like he was in the photo. So, it's really important to document small things that you may find along the way in an investigation. They may seem irrelevant now, but they could turn out to have much more of an impact later on down the road.

Google Images

Google Images is not just for searching images. Of course, you can type a keyword or a person's name and marvel at the appearance of all the actual matches, partial matches, and matches that have absolutely nothing to do with your original search term.

But . . . Did you know Google Images also allows you to reverse image search an existing image? To do this, click the camera icon shown in Figure 15.7.



Figure 15.7

Clicking this icon will allow you to either paste a specific image URL or upload an image directly. Figure 15.8 shows the image upload screen.

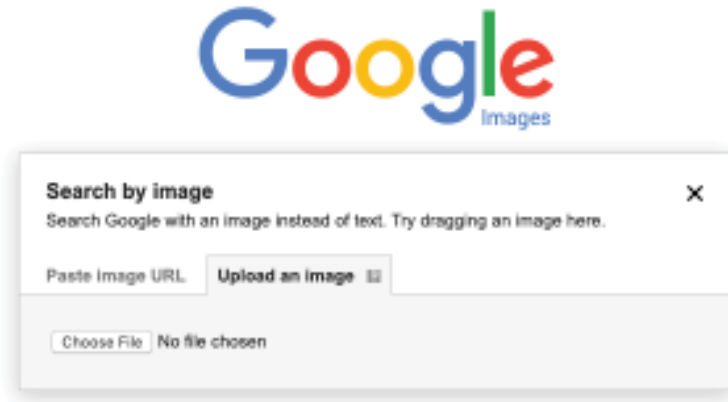


Figure 15.8

WARNING It is worth noting that I have never, not one single time, received a useful match using a Google Images reverse search. I am always optimistic, though... and for the sake of thoroughness, I always try here first.

Searching for Gold

A while back, someone loosely associated with TDO sent me a few pictures he claimed were associated with the threat actor known as Tessa88. Apparently Tessa sent a number of photos to this threat actor in some attempt to either seem cool, or verify his identity. Figure 15.9 is one of those photos, which shows what appear to be stolen credit cards.

You never know where or why this image might show up elsewhere, but it is important to leave no stone unturned, which is why I like to start my search in Google Images. The results of the image search are shown in Figure 15.10.

We have a match!

I swear, these results were completely unexpected. I am blown away that we have a match—and even more so that we have a useful one. Sometimes you get lucky. The single page that matches the picture is blackforum.cc, a Russian carding forum. Let's explore.

Following the Trail

Following the trail to the Russian carding forum, we can see a post where the cards are listed for sale by user "Bankir" (Figure 15.11).

This is a really solid hit—I already had the other two images in this picture. This tells me that the person who posted this message is either the same person or working with the same group.



Figure 15.9

In either case, we have learned a new username he is associated with: “Bankir.” We also now have his telegram account that we can use if we choose to engage with him. File that one under “win.”

TinEye

TinEye (www.tineye.com) is very similar to the Google Image reverse search and by “very similar,” I mean *exactly* like Google Image reverse search. TinEye is hit-or-miss, just like Google, so you really should try them both.

Figure 15.12 shows the TinEye search page.

For completeness, let’s search TinEye for the same credit card image in Figure 15.9. Unfortunately, Figure 15.13 shows that TinEye was unable to match the image found by Google Images.

Let’s run another search. This time, we’re checking for matches on one of Cyper’s old profile avatars (Figure 15.14).

This is a fairly common image, and a good way to illustrate the difference between TinEye and Google Images. Running this image through the Google Image reverse search returns roughly 20 million results.

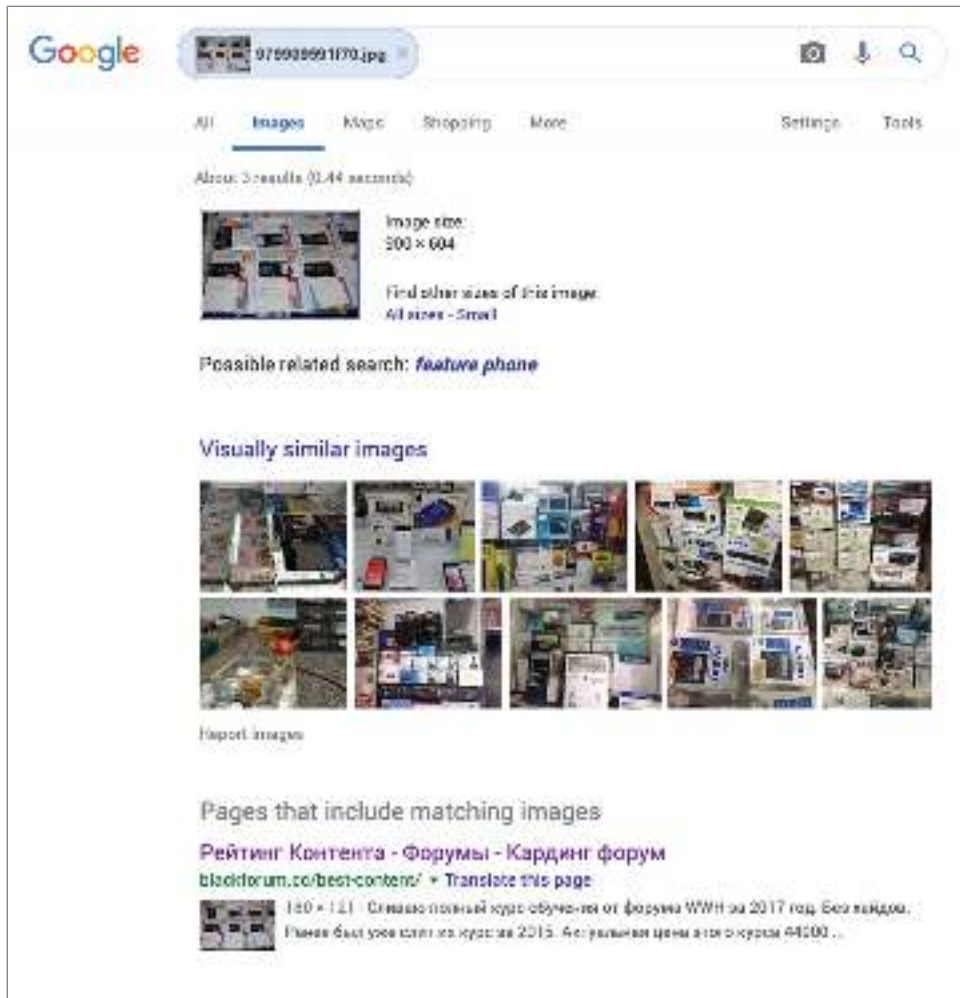


Figure 15.10

On the other hand, TinEye returns a more manageable number (Figure 15.15).

The first hit on this image is a profile avatar for a user on antichat.ru, an old-school Russian hacking forum. This is a solid match, and absolutely worth investigating. It is also another great example of why you should never rely on just one tool for all of your results.

Interestingly enough, if you look through the search results, you will eventually find Cyper's protected Twitter page (Figure 15.16).

Buy a bank debit card of Sberbank, VTB-BK, Alfabank | Sale site ...

At the moment Bank is 23 March 2020 11:00

Sale of debit cards of banks of the Russian Federation

222 questions about the acquisition and the use of the LSPK

The kit includes:

1. Application to the card
2. Debit card
3. The code for withdrawing from an ATM
4. Simka linked to the card
5. Internet access Banking
6. Code word (and / or digital code)
7. Color scan of passport passport

Warranty:
When using a card for white threats, for receiving white threats and Mark / gray list, washed through bitcoins and exchangers, the card will live for the whole life. The kit immediately.

Availability and Cost:
Sberbank
Categories and Prices:
Movements - 4000r
Classic - 5500r
Gold - 8000r

Alfa Bank Debit Cards
Categories and Prices:
Classic - 2000r
Gold - 4000r

Sberbank VTB-BK Bank
Categories and Prices:
Classic - 5500r
Gold - 8000r

Promosbank Debit Cards
Categories and Prices:
Classic - 5500r
Gold - 8000r

Delivery terms: Moscow and St. Petersburg delivery by courier. To the regions by an anonymous courier service (DHL, DHL) on presentation of a passport or other documents to



Write in the LAN or Telegram - @BankerSeller

Figure 15.11

 TinEye

Technology Products Log In

Reverse Image Search

Search by image and find where that image appears online

 Upload or enter image URL 

[How to use TinEye](#)

Figure 15.12



Figure 15.13



Figure 15.14

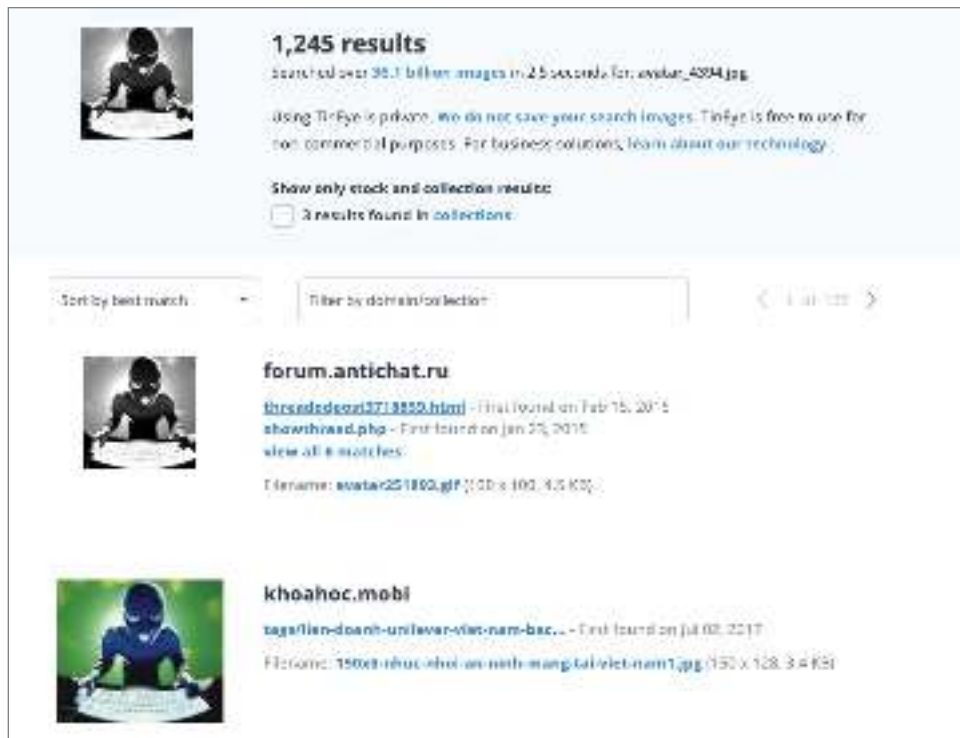


Figure 15.15

Another excellent match. I have found that threat actors like to reuse profile pictures, so finding a new Twitter or social media page associated with a threat actor can often come down to something simple like matching a profile image. And hey, look at that! I had no idea that he follows me! That's kind of funny, but it also confirms one very important detail: this account is still active.



Figure 15.16

EagleEye

EagleEye is an image search tool designed to reverse-search social media profile pictures. The main drawback of EagleEye is that you need to know a part of the person's name in order to search for more matches of their profile pictures. This may not necessarily be a problem if you already know the name of the threat actor or target and are just looking for more information on that person.

EagleEye first searches the name across Facebook, then performs a reverse image search on the picture across Google, ImageRaider, and Yandex to find the person's other social media accounts. The end result is a nicely formatted PDF report.

Another good use of this tool would be to check if someone is potentially catfishing you. It's always good to run someone's picture through social media and see what other matches you can come back with.

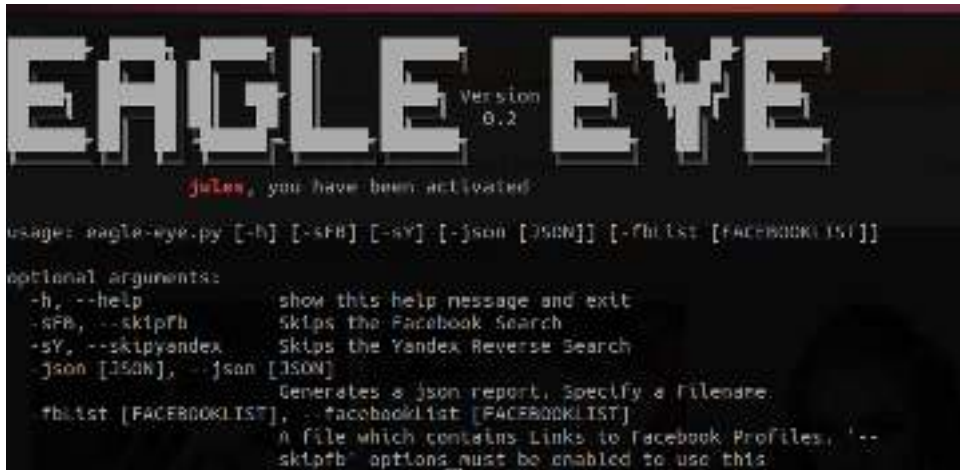
You can download EagleEye at <https://github.com/ThoughtfulDev/EagleEye>.

Figure 15.17 shows the main EagleEye screen.

Searching for Images

For our example image, let's search for one of my press photos (Figure 15.18) and see what kind of results we get back.

Figure 15.19 shows what the typical EagleEye output looks like when processing an image.



```

EAGLE EYE Version 0.2
jules, you have been activated

usage: eagle-eye.py [-h] [-sfb] [-sy] [-json [JSON]] [-fblist [FACEBOOKLIST]]

optional arguments:
  -h, --help            show this help message and exit
  -sfb, --skipfb         skips the Facebook Search
  -sy, --skipyandex     skips the Yandex Reverse Search
  -json [JSON], --json [JSON]
                        Generates a json report. Specify a filename.
  -fblist [FACEBOOKLIST], --facebooklist [FACEBOOKLIST]
                        a file which contains links to Facebook Profiles. '--skipfb' options must be enabled to use this

```

Figure 15.17



Figure 15.18

In the background, EagleEye loads a webdriver and automatically searches Google for your image. Figure 15.20 shows the automatic reverse Google search process.

While not super useful, EagleEye was at least able to identify my Facebook page by searching for accounts using the information gathered from my profile picture (Figure 15.21).

Not bad, actually. It found a few of my profiles, so there is definitely some merit to this tool and worth trying if you have pictures of people you want to identify.

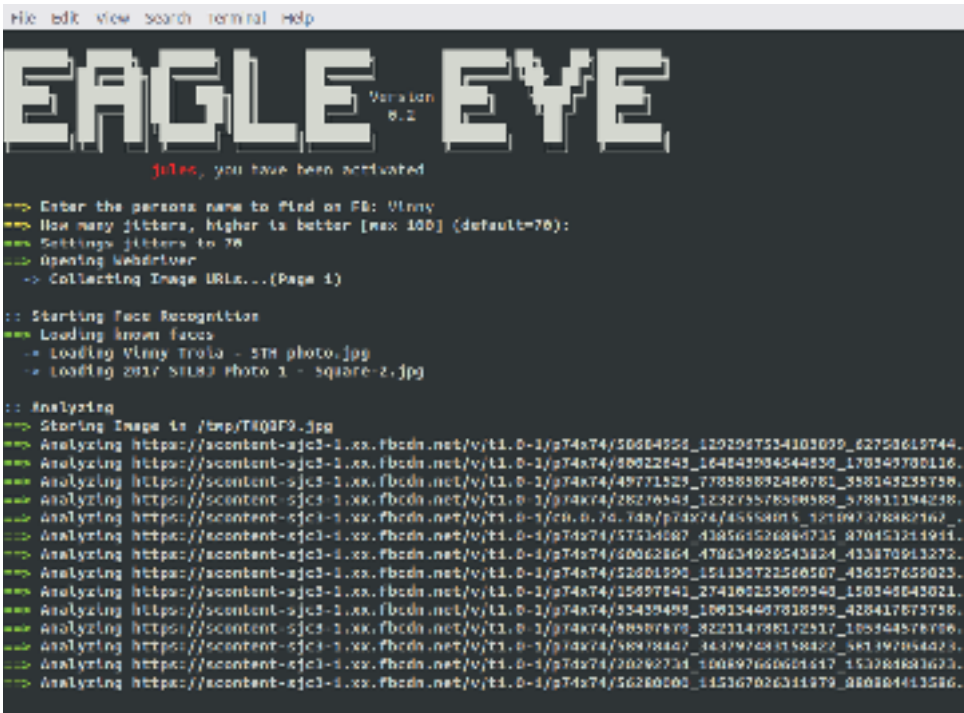


Figure 15.19



Figure 15.20

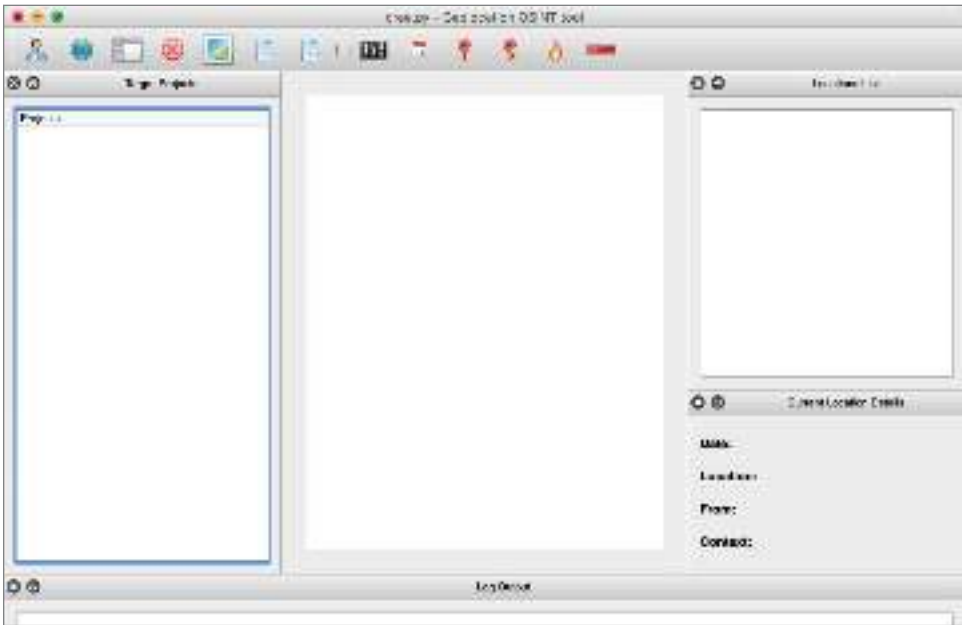


Figure 15.22

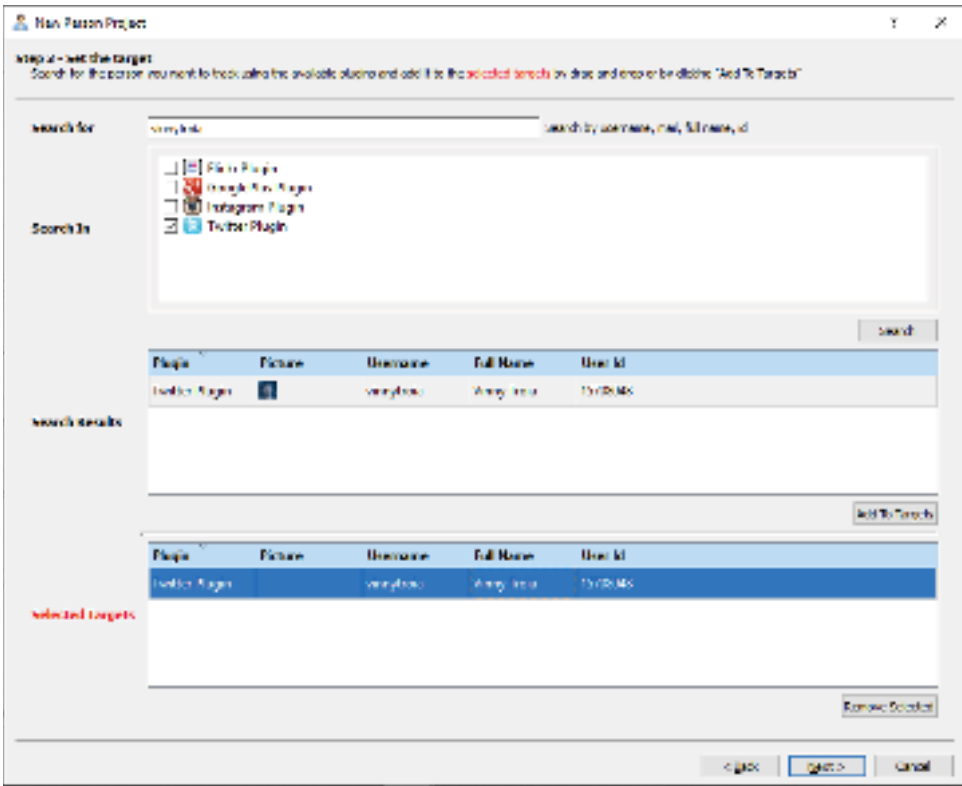


Figure 15.23

Entering my name in the “Search for” box will populate the Selected Targets area if a successful match is found (or if multiple potential matches are found). After selecting your target, click the Next button to begin analyzing the user’s tweets.

After the initial analysis is complete, the target’s Twitter profile information is displayed. At this point, you should look out for one of the following two lines:

- User has enabled the possibility to geolocate their tweets
- User has disabled the possibility to geolocate their tweets

If the user’s tweets have geolocation information, Cree.py will display a list of the target’s tweets and their respective location information. Looking at the results for my Twitter account (Figure 15.24), we can see seven tweets that have geolocation information.

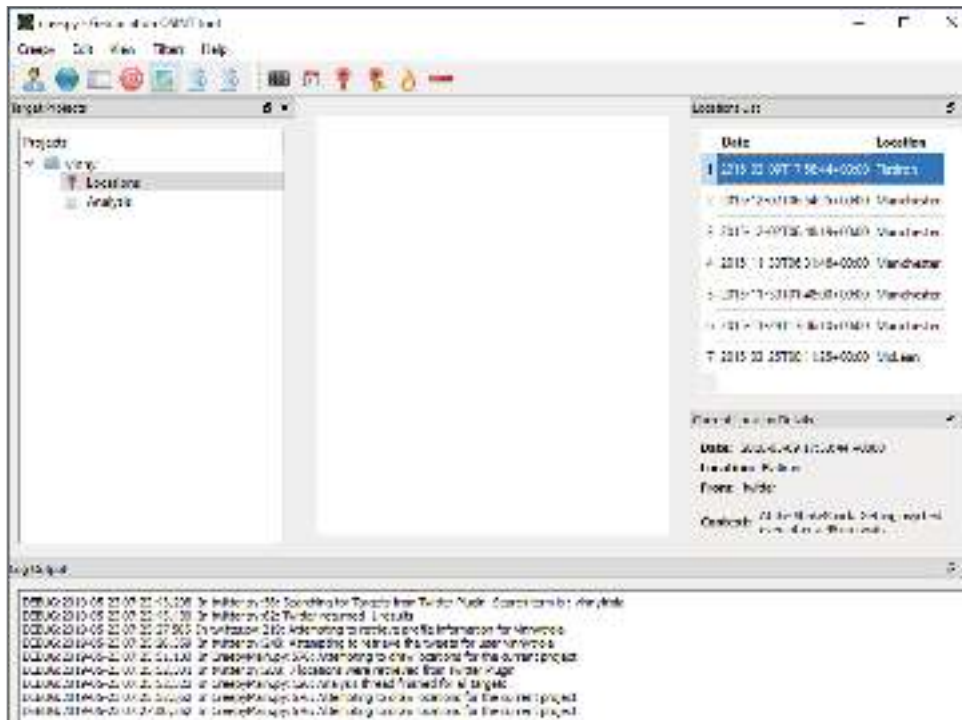


Figure 15.24

At this point, you can filter locations by date or their positions on the map. In addition, you can set up a custom date filter to look for tweets within specific date ranges, or filter out inaccurate locations. Each time you update the location filter, the map is automatically refreshed.

You can also export your results as a CSV, which is very useful for keeping a record of the information and having something that you can index and search later on in your investigation.

Table 15.1 is a CSV export of my tweets with enabled geolocation information.

Table 15.1: CSV Export with Geolocation Information

TIMESTAMP	LATITUDE	LONGITUDE	LOCATION NAME	CONTEXT
2016-03-09 17:58:44 +0000	40.7393845	-73.990098	Flatiron	At the ShakeShack. Getting psyched even after a 45 min wait.
2015-12-02 06:54:25 +0000	38.577462	-90.499221	Manchester	ICYMI - 128GB USB3 Flash Drive - \$25 - https://t.co/ LhLWPotjVz
2015-12-02 06:48:19 +0000	38.577462	-90.499221	Manchester	Wow. 128GB USB3 flash drive only \$25. SSD prices are dropping like crazy.
2015-11-30 06:31:46 +0000	38.577462	-90.499221	Manchester	has anyone started shopping yet? is there anything good out there worth buying right now? #BlackFriday
2015-11-30 01:48:00 +0000	38.577462	-90.499221	Manchester	Ok, look. I am as big a fan of #WalkingDead as anyone, but at some point shouldn't the zombies all collapse under their own weight?
2015-03-25 00:11:25 +0000	38.9374855	-77.203885	McLean	Hello, Langley.

Looking for geolocation data in a threat actor's tweets seems like a long shot, but you should never underestimate a cyber criminal's ability to leave a sloppy trail like this. There are plenty of stories about criminals getting arrested for a major crime like transporting drugs because they failed to stop at a stop sign. This is no different.

IP Address Tracking

At this year's Derbycon, I saw John Strand give a talk on cyber attribution. In his talk, he demonstrated a way to get better results when attempting to pinpoint the physical location of an IP address.

EXPERT TIP: JOHN STRAND

I was testing Canarytokens on a user who I knew was in downtown Louisville, KY. When the user opened the fake malicious package, I was sent back his IP address. When I performed an geo-location lookup on his IP address lookup, and the result was not very accurate – the results came back with a location that was about 20 miles away.

I have found that one way to get better granularity on your geo-location lookup is to use the traceroute command on the IP address in question. Then, when looking at your results, take the IP address of the *last routing hop* and geo-locate that IP address instead.

This will get you *much* closer. In my test case, I was able to pin-point the IP address to within one block of the hotel.

The reason why this works is because most IP addresses (for home users in particular) are handed out via random DHCP. The IPs can bounce around and a geo-ip lookup will have no way of knowing where you actually are.

But using routing infrastructure from the traceroute command, we can typically get much closer and much more accurate results for a location.

Summary

This first chapter in the “People Hunting” section covered a number of tools that can be used to help track people online, including reverse image searching, background checks, various people lookup tools, and ways to gather geolocation information from social media.

The next chapter will dive deeper into researching social media profiles, which will include tools and techniques that can help identify multiple aliases, accounts, and phone numbers associated with a target.

Searching Social Media

This chapter focuses on tools and techniques that will help in your discovery of a target's alternate aliases and social media profiles. We will look at different ways to search and find names, aliases, and phone numbers in order to eventually build a threat actor tracking matrix in Chapter 17.

Several examples in this chapter will focus on Nathan Wyatt, who is currently fighting U.S. extradition for crimes associated with The Dark Overlord.

My official position on Wyatt is that I believe he used the alias Arnie, and that the group used him as their patsy. This theory is supported by the following conversation I had with Columbine (who I believe to be a close associate of NSA@rows.io).

Columbine: Yeah I dont know where the real Revolt is now but idc tbh

Columbine: We never spoke that much anyway

Columbine: A lot of the other n***** gone too

Columbine: Like NSA@rows.io

Columbine: They formed thedarkoverload

Columbine: that ransoming group

SoundCard: yeah i know about them well

Columbine: I did hear that Arnie was getting mad at NSA though

SoundCard: i will say that they did a good job with marketing

Columbine: So internal tension between the group

Columbine: Yeah
Columbine: They were really good with that
SoundCard: The drama is funny.
SoundCard: But ok, I'll bite. why?
Columbine: NSA barely did s*** for TDO I heard
Columbine: and still he got the money
Columbine: so Arnie (who did most of the work) got mad
Columbine: Thats my understanding of it
SoundCard: oh see, i thought Arnie was that guy who was put in jail
Columbine: Nah they're still around
Columbine: Unless they mad a critical mistake I doubt they'd be arrested
Columbine: Which I dont think so
Columbine: Out of all them I expect NSA to get arrested
SoundCard: it was a uk guy
Columbine: Yeah you heard his call?
SoundCard: hahahahah yeah
SoundCard: what a d***
Columbine: Yeah ikr

If I am correct, then this conversation makes it clear that Columbine is trying to shift all of the major TDO hacks to Arnie and away from NSA (A core head of TDO).

The important takeaway here is that not everything you gather will be valid or legitimate. Many of the skilled threat actors will leave clues designed to specifically knock you off their trail.

When gathering information, especially on social media, you should always ask yourself whether what you have found is valid, or if it was placed there by someone wanting you to find it.

Question everything.

OSINT.rest

OSINT.rest is a website from the creators of the SocialLinks plugin for Maltego (<https://www.mtg-bi.com>). SocialLinks is *by far* my favorite Maltego plugin—I was initially planning on covering Maltego just to talk about its plugin, but now that there is a separate website that allows you to collect the same data, I like this approach better.

OSINT.rest (www.osint.rest) is a giant API that allows you to collect social media information on your targets. You can query the API using CURL commands in the command line, or use a tool like Postman (<https://www.getpostman.com>). This is a premium service—you can purchase API keys and queries in bulk.

OSINT.rest allows API searches for the following:

- Search for user by phone or email
- Facebook (photos, pages, video, users, posts, likes, places)
- Instagram (search by alias)
- Image search
- Twitter search
- Foursquare
- SL DB
- VK.com
- MySpace

When viewing the OSINT.rest API docs, you'll find a button that says "Run in Postman." Click this button to load the API directly into your Postman app (currently available for Windows or Mac).

The menu on the left will be populated with searches from OSINT.rest. Figure 16.1 shows the Postman app with the OSINT.rest API connection loaded, and the "Search Information by Phone" option highlighted.



Figure 16.1

For our trial, I put in my cell phone number. The window in Figure 16.2 shows the successful results from this search.

This API only works with U.S. numbers, but the results are obviously very powerful if you need help in reverse searching a phone number.

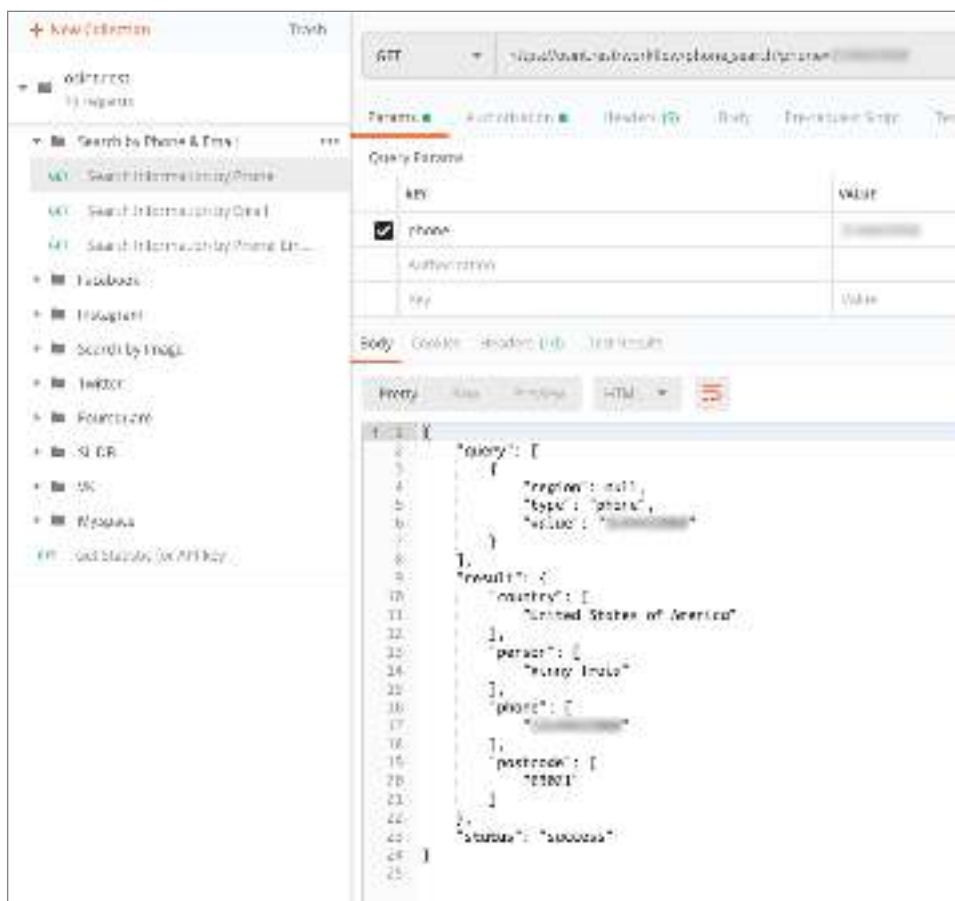


Figure 16.2

OSINT.rest has nearly a hundred different options for searching Facebook, which include the ability to search for users, a user's friends, people who comment on posts, videos, pages, and much more.

For our first search, let's use the profile of Nathan Wyatt, someone who was convicted and sent to prison in 2017 for hacking the phone of Pippa Middleton. As of 2019, Nathan Wyatt is fighting extradition to the U.S. for crimes associated with The Dark Overlord (<https://www.dailymail.co.uk/news/article-6578213/Stay-home-father-accused-hacking-fights-extradition-US.html>).

A simple search on Facebook will reveal his public Facebook profile. Figure 16.3 shows his Facebook page and a profile ID of 100010064775327 in the URL bar.

Having the target's Facebook ID saves us a step of having to look for it. Most searches require the exact ID, so you will eventually need to find it. Let's dig into the Facebook capabilities of OSINT.rest.

Knowing who your target is connected to can be a very easy way to identify other threat actors or others with mutual connections. Getting a JSON list of someone's friends is as simple as entering their Facebook ID in the query parameters.

Figure 16.4 shows the output of this query.



Figure 16.3

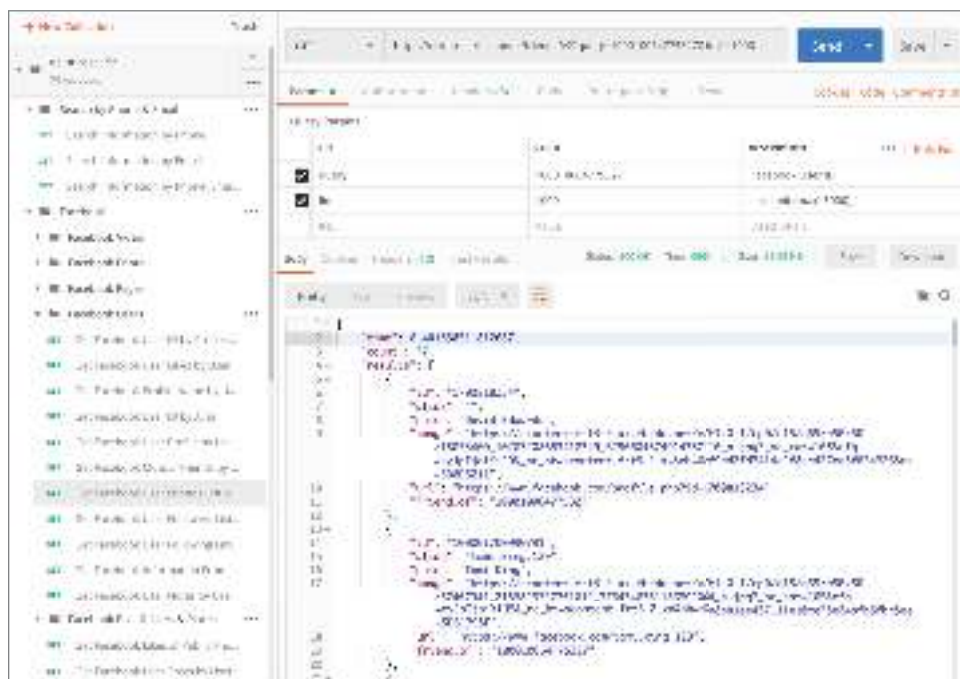


Figure 16.4

NOTE Here is a very specific example of why knowing someone’s friends can be important. A while back I was looking for a specific threat actor (I knew his real name). I was looking through his friends list and there were a few people that caught my eye. After looking at their Facebook profiles and friends lists, I noticed that they were actually friends with my target’s alternate Facebook accounts! The photo was an immediate giveaway, but since the name was different, I never would have found that profile if I had not taken the time to look through his friends list.

We can see from the output that Nathan Wyatt has 77 public Facebook friends. This output also gives us the names, IDs, images, and direct URLs of his friends. This is one particular area where using Maltego makes this process much easier since the app automatically creates a visual of all the connections for you, which I demonstrate later in this section.

We know the alias “CraftyCockney” is directly associated with Nathan Wyatt from all the public news stories and court reports, so that is a great place to start searching.

For this search, let’s try Instagram. Using the Instagram API option on the left, Figure 16.5 shows a query of “Craftycockney” next to the results.

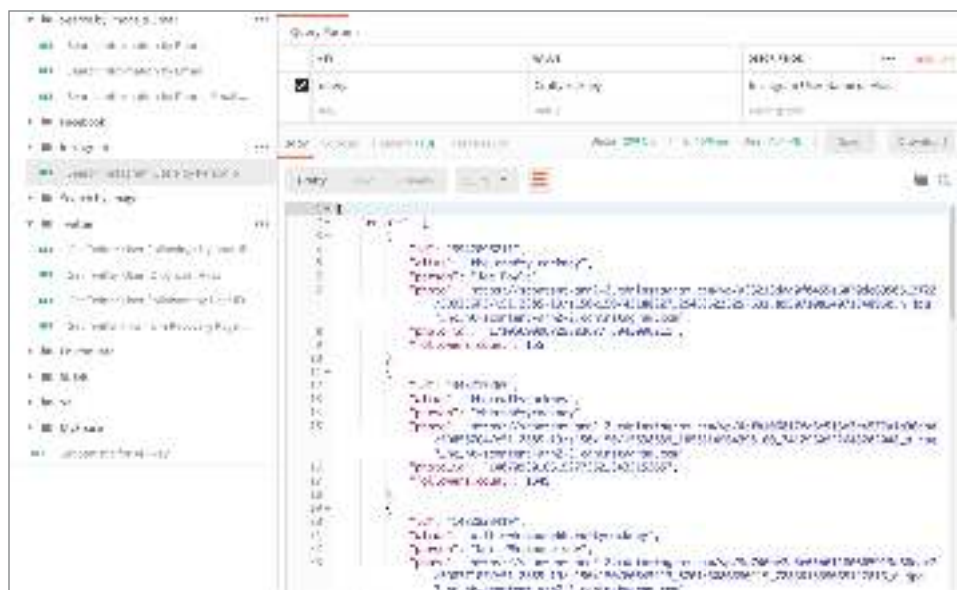


Figure 16.5

This API search has returned everyone on Instagram with CraftyCockney as part of their username. We have only about 20 results, so looking through all of these will be worth the time to identify the right target profile.

If you have other aliases to look up, just rinse and repeat. Let’s try one more.

Another Test Subject

For this next example, let's look up our good friend and fellow security expert Christopher Meunier (aka WhitePacket). Most of Chris's Facebook profiles are offline, but due to the magic of Facebook's search algorithms, I typed in his name and out popped a new profile with his profile picture and an intentionally misspelled name. The Facebook profile was listed as Chris Maunier, but for some reason the search results displayed the profile anyway. Maybe Facebook had associated his photo, or maybe it was just giving me close matches. Either way, score!

In order to perform any searches, we need to know his Facebook ID. We can tell from the URL that his username is christopher.maunier.923. All we need to do is enter that username in the "Get Facebook User ID by Profile" API search (Figure 16.6).

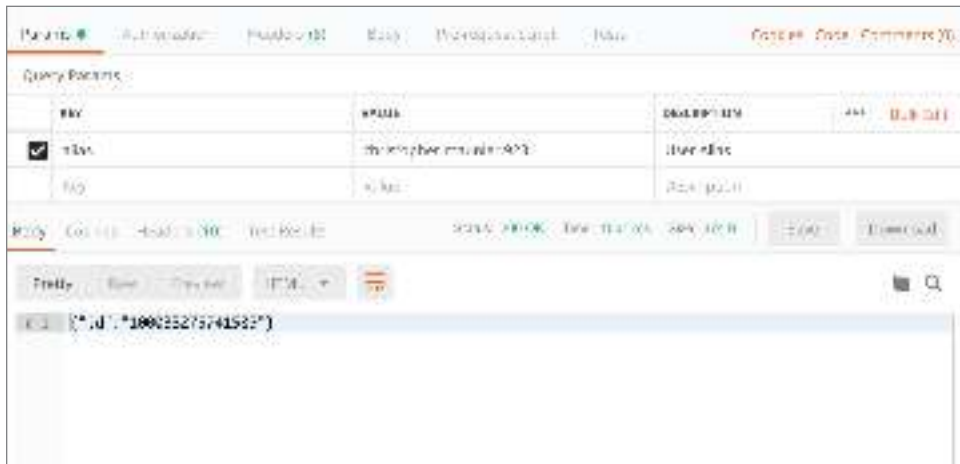


Figure 16.6

Once we run the search, we easily obtain his Facebook ID of 100033275741589.

Unfortunately, Chris's profile is mostly private, but there is still a decent amount of information we can get from it. To start, "Get Facebook User Profile by User ID" (Figure 16.7) will return a summary of the target's Facebook profile, which includes recent posts, headline, full name, direct URL, photo URL, and more.

One thing we can immediately see from these results is the URL of his Facebook photo. Lucky for us, OSINT.rest also includes a reverse image search API. Let's first start by navigating to this URL. The public post and profile photo for user 100033275741589 (Figure 16.8) contains the caption "Fak u."

Now that we have a profile photo, we can save it for later or copy the URL and perform a reverse image search.

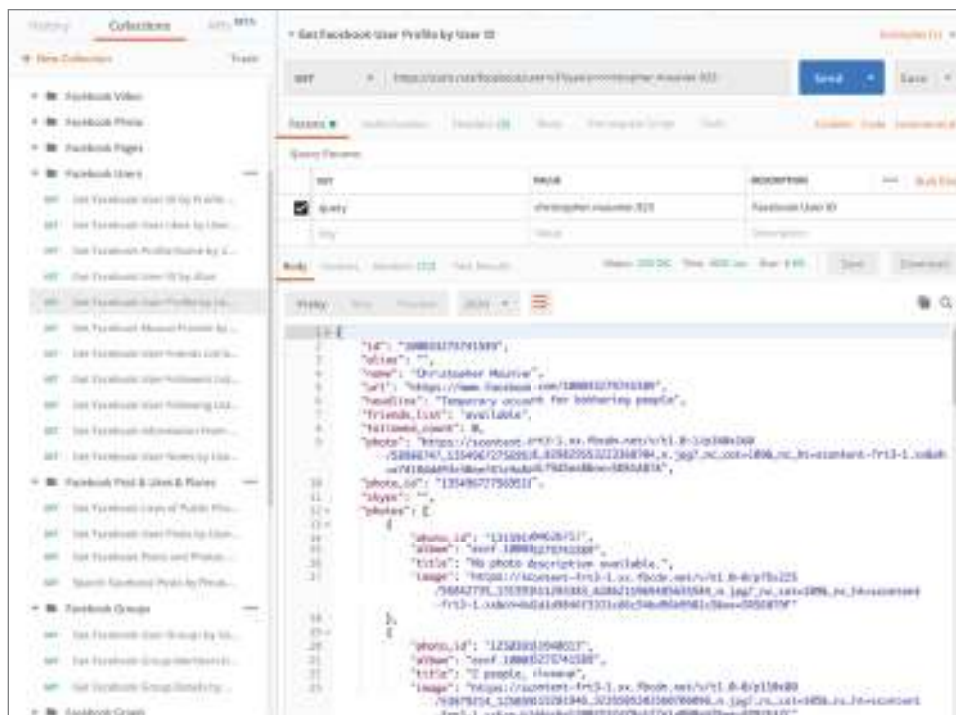


Figure 16.7



Figure 16.8

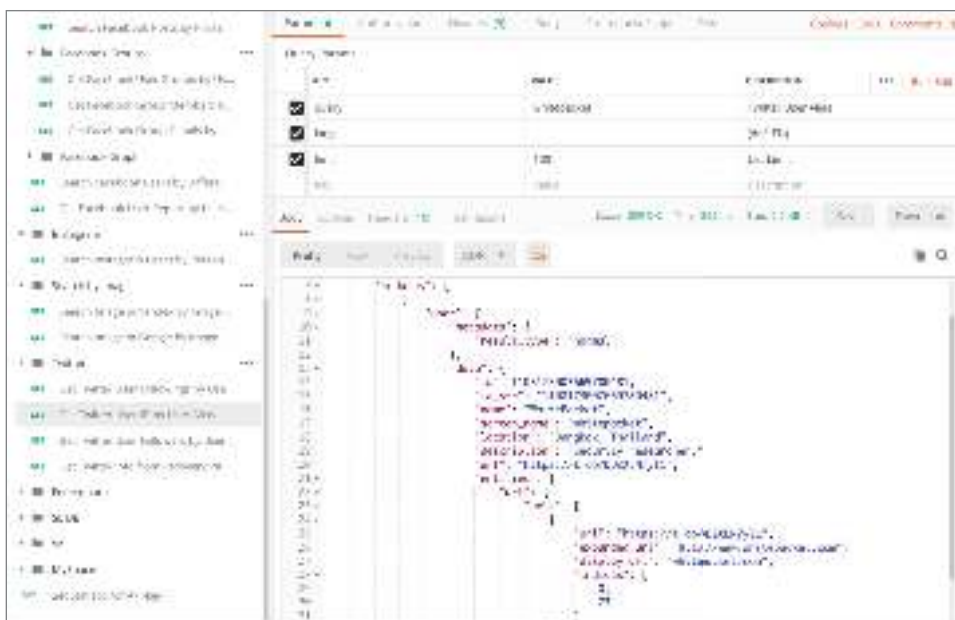


Figure 16.10

We have a match—which is actually surprising because WhitePacket shut down his Twitter account a few months ago. I guess he decided to start it back up.

Now that we have his Twitter ID, we can perform all sorts of fun and interesting searches, including getting lists of who he is following and his followers. Fortunately, this account is fairly new, so he does not have that many followers. In my mind, that means the people he is following (or that are following him) are most likely quality contacts worth investigating.

Figure 16.11 shows a search to get a list of Twitter followers based on Twitter ID.

It's that simple. If he were a target, we could then expand our search to also get lists of tweets that he liked or retweeted, and use that to assemble a threat actor matrix.

We only covered a handful of the different search APIs available in OSINT.rest, so I highly recommend checking out this service.

SocialLinks: For Maltego Users

Many, many other books have (exhaustively) covered Maltego, so I made the decision not to include it in this book. I personally can't live without it, but I thought it would be more interesting and beneficial to focus on lesser-known tools and techniques that can potentially provide more value to an investigation. That being said, since this section is discussing the OSINT.rest API, I do want to point out the awesome capabilities of the SocialLinks plugin (which is essentially a GUI version of OSINT.rest).

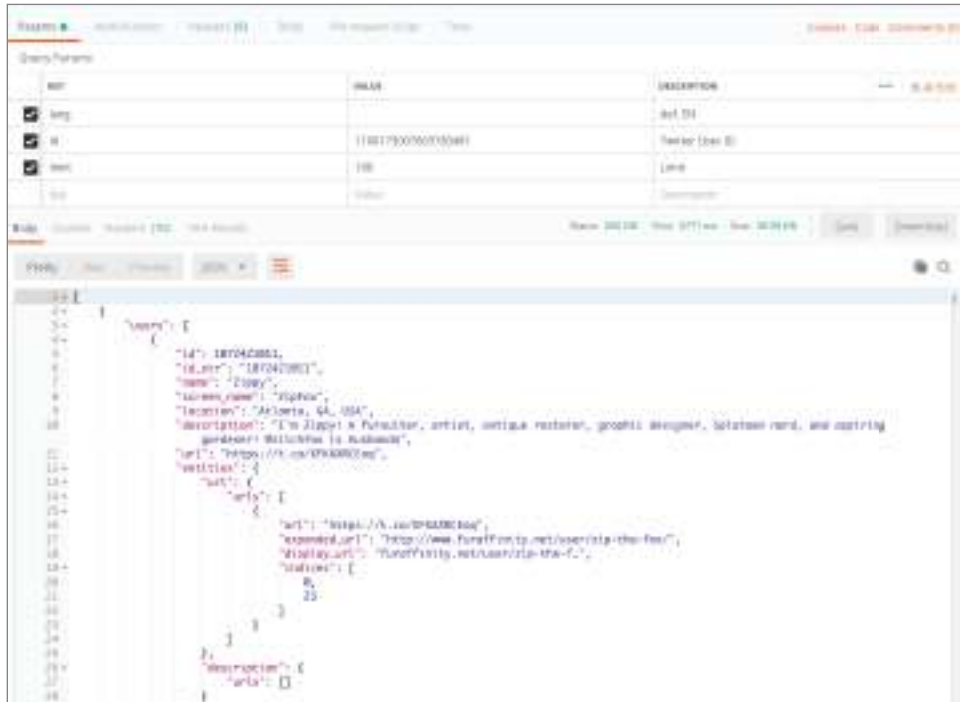


Figure 16.11

To illustrate two examples of the awesomely useful potential of this plugin, let's run a search against the email address `hackernike@live.ca`. SocialLinks (using the same API as OSINT.rest) will search for that email address and automatically create visual connections based on any information it finds. Figure 16.12 shows the connections associated with our target email automatically created using the SocialLinks plugin.

Just from running a single search, we can see two different MySpace pages, four different Skype accounts, two IP addresses, and a fake name. That is amazingly powerful stuff!

For our final example, Figure 16.13 shows the results of using the SocialLinks plugin within Maltego to search for WhitePacket's public email address, `chris@whitepacket.com`.

Again, with just one single search, we can see so much information: two Skype profiles, a few GitHub links (one for WhitePacket's published ZIB Trojan and ZLO Botnet), a Google account, and a Foursquare page! Amazing.

Skiptracer

Skiptracer is a Python tool that uses basic web scraping to help compile passive information on a target. Skiptracer has a number of modules that harvest information based on telephone numbers, email addresses, screen names, real names (first and last names), home addresses, IPs, hostnames, license plate numbers, and breached website credentials (using HIBP.com).

The email modules will search for associated accounts based on your target email address on sites such as LinkedIn, HaveIBeenPwned, MySpace, and AdvancedBackgroundChecks.

The name modules will search for a target's first and last name on sites like Truth Finder, True People, and AdvancedBackgroundChecks.

The phone modules look up U.S.-based phone numbers (including performing a reverse search) on TruePeopleSearch, Who Called, and 411.com. The license plate module will search a nationwide database of registered U.S. license plates.

Finally, the screenname (username) module will search for similar usernames on sites such as Knowem and Namechk.

You can download Skiptracer for free at <https://github.com/xillwillx/skiptracer>.

Running a Search

After launching Skiptracer, you will see a lookup menu (shown in Figure 16.14). From there, you can select which module to use.

Searching for an Email Address

Let's start off by searching for an email address. Pressing 1 on your keyboard while on the home menu will bring you to the email search menu (Figure 16.15).

Next, we can enter in any target email. Let's go back to a previous example and use `hackernike@live.ca`. For some reason, searching LinkedIn does not work, which means you can't search "All." Instead, we have to run each module individually. For this example, let's run a search for that email address using option 3, which will search the email against Troy Hunt's Have I Been Pwned (HIBP) database (Figure 16.16).

```

root@Kunar: /opt/skiptraces: python skiptraces.py

SKIPTRACES
-0x0-

iIMOB
https://i11mob.org

SKIPTRACES
-0x0-

iIMOB
https://i11mob.org

[?] Invalid menu:
[1] Email - Search targets by email address
[2] Name - Search targets by First Last name combination
[3] Phone - Search targets by telephone number
[4] ScreenName - Search targets by known alias
[5] Plate - Search targets by license plate
[6] Domain - Search targets by Domain
[7] Help - Details the application and user roles
[8] Report - Generates a docx report from queries
[9] Exit - Terminate the application
[?] Select a number to continue:

```

Figure 16.14

```

[?] 8-Mail search menu: Target info
[1] All - Run all modules associated to the email module group
[2] LinkedIn - Check if user exposes information through LinkedIn
[3] HaveIBeenPwned - Check email against known compromised networks
[4] Myspace - Check if users account has a registered account
[5] AdvancedBackgroundChecks - Run email through public page of paid access
[6] Reset Target - Reset the Email to new target address
[7] Back - Return to main menu
[?] Select a number to continue:

```

Figure 16.15

```

[?] 8-Mail search menu: Target info - hockernike@live.cn
[1] All - Run all modules associated to the email module group
[2] LinkedIn - Check if user exposes information through LinkedIn
[3] HaveIBeenPwned - Check email against known compromised networks
[4] Myspace - Check if users account has a registered account
[5] AdvancedBackgroundChecks - Run email through public page of paid access
[6] Reset Target - Reset the Email to new target address
[7] Back - Return to main menu
[?] Select a number to continue:

```

Figure 16.16

Here are the results of the search against HIBP:

```

[?] HaveIBeenPwned
[+] Dump Name: 000webhost
[=] Domain: 000webhost.com
[=] Breach: 2015-03-01

```

```

[=] Exposes:
  [-] DataSet: Email addresses
  [-] DataSet: IP addresses
  [-] DataSet: Names
  [-] DataSet: Passwords
[+] Dump Name: Adobe
[=] Domain: adobe.com
[=] Breach: 2013-10-04
[=] Exposes:
  [-] DataSet: Email addresses
  [-] DataSet: Password hints
  [-] DataSet: Passwords
  [-] DataSet: Usernames
[+] Dump Name: CashCrate
[=] Domain: cashcrate.com
[=] Breach: 2016-11-17
[=] Exposes:
  [-] DataSet: Email addresses
  [-] DataSet: Names
  [-] DataSet: Passwords
  [-] DataSet: Physical addresses
[+] Dump Name: DaniWeb
[=] Domain: daniweb.com
[=] Breach: 2015-12-01
[=] Exposes:
  [-] DataSet: Email addresses
  [-] DataSet: IP addresses
  [-] DataSet: Passwords
[+] Dump Name: MySpace
[=] Domain: myspace.com
[=] Breach: 2008-07-01
[=] Exposes:
  [-] DataSet: Email addresses
  [-] DataSet: Passwords
  [-] DataSet: Usernames
[+] Dump Name: xat
[=] Domain: xat.com
[=] Breach: 2015-11-04
[=] Exposes:
  [-] DataSet: Email addresses
  [-] DataSet: IP addresses
  [-] DataSet: Passwords
  [-] DataSet: Usernames
  [-] DataSet: Website activity

```

Even though HIBP won't show us any of the compromised information, we can still gain a lot of information just in knowing *where* a particular email is used. This will come in especially handy in Chapter 17 when we discuss building a threat actor matrix, and using a site's password resets to build additional clues

about our targets. In the meantime, every one of these sites has information on our target—sometimes even public information available on forums or social media pages.

Searching for a Phone Number

I am still surprised at how much information is returned when you search for a phone number in Skiptracer. Figure 16.17 shows a search against my personal cell phone. I blocked out the number, but I am using this as an example because I want to comment on the results.

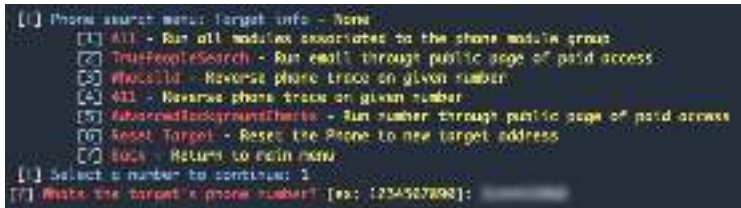


Figure 16.17

The following (extremely verbose) results are displayed when searching for my cell phone number. I will not display any of my personal information, but I feel it would be really useful and informative to show the type of information being displayed, and discuss whether it is valid.

I have included my notes in [brackets] next to each result:

```
[+] Alias:
[=] AKA: Toria Vinney
[=] AKA: Vinny Troia
[=] AKA: Vinnie Troia
[=] AKA: Vincenzo Troia
[+] Related:
[+] Associate(s):
[=] Known Associate: *** [Someone I haven't spoken to in 15 years]
[=] Known Associate: *** [Removed]
[=] Known Associate: *** [Never heard of this person]
[=] Known Associate: *** [Never heard of this person]
[=] Known Associate: *** [My next door neighbor]
[=] Known Associate: *** [Used to live at my address]
[=] Known Associate: *** [Never heard of this person]
[=] Known Associate: *** [Family member]
[=] Known Associate: *** [Aunt]
[=] Known Associate: *** [Removed]
[=] Known Associate: *** [Wife's ex]
[=] Known Associate: *** [Removed]
[=] Known Associate: *** [Removed]
```



```
[+] Addresses.:
  [=] Current Address:
    [-] Street: ** [Accurate]
    [-] City: ** [Accurate]
    [-] State: ** [Accurate]
    [-] ZipCode: ** [Accurate]
  [=] Prev. Address:
    [-] Street: ** [Accurate]
    [-] City: ** [Accurate]
    [-] State: ** [Accurate]
    [-] ZipCode: ** [Accurate]
[+] Name: ** [Accurate]
[+] Phone:
  [=] #: ** [Accurate]
  [=] #: ** [Can't remember]
  [=] #: ** [Can't remember]
[+] Email:
[+] Addresses.:
  [=] ** [Removed]
[+] Related:
  [=] Known Relative: ** [Removed Accurate]
  [=] Known Relative: ** [Removed Accurate]
  [=] Known Relative: ** [Removed Wrong]
```

Regardless of how many results were valid or invalid, this is still a fairly exhaustive list that was derived from a single search!

Searching Usernames

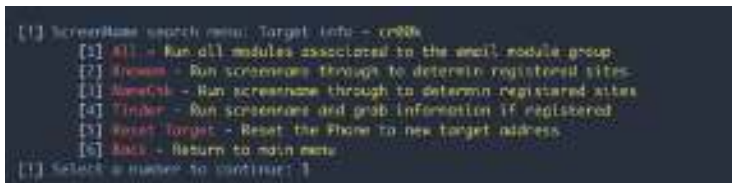
The username search will check lists of known sites to see if a URL exists with the username on a particular site. The drawback is that the app is not actually checking if the user exists. It is displaying a positive match if there is a URL that comes up. That means if the site has a 404 page, or a redirect, there will be a positive match. Many other tools (like SpiderFoot and Intrigue.io) provide similar searches. Wherever you decide to run this type of search, this is absolutely something you have to do for *every* username you come across. A single positive match can open up a new world of research for you, and you can't discount the fact that threat actors will reuse usernames on multiple sites.

Figure 16.18 shows a search for user cr00k.

NOTE I haven't talked much about Cr00k yet (more on him in the coming chapters), but he/his username was associated with TDO as one of their first data peddlers. His name was all over the hacker forums, and as a result, all over any TDO-related threat reports. Interestingly enough, the real person behind cr00k was smart enough to steal another carder's username, which is a common tactic of his.

This is a very smart move, and something to watch out for. When people researched the name cr00k, they would inevitably be led down a path of misinformation put in place by other threat actors using the same name.

Not only that, but switching and reusing names within members of the same group is a very common tactic used to create confusion. You should definitely keep this in mind when dealing with a long timeline of events.



```
[1] ScreenName search menu: Target info - cr00k
[+] [1] - Run all modules associated to the email module group
[+] [2] known - Run screenname through to determine registered sites
[+] [3] Namechk - Run screenname through to determine registered sites
[+] [4] Tinder - Run screenname and grab information if registered
[+] [5] Reset Target - Reset the phone to new target address
[+] [6] Back - Return to main menu
[+] [7] Select a number to continue: ]
```

Figure 16.18

The username search for cr00k displays the following results:

```
[?] Whats the target's screenname? [ex: (AcIdBurn|Zer0C001)]: cr00k

[?] Knowem
[+] Account: Blogger
[+] Account: DailyMotion
[+] Account: facebook
[+] Account: foursquare
[+] Account: Imgur
[+] Account: LinkedIn
[+] Account: MySpace
[+] Account: Pinterest
[+] Account: reddit
[+] Account: Tumblr
[+] Account: Twitter
[+] Account: Typepad
[+] Account: Wordpress
[+] Account: YouTube

[?] Namechk
[+] Acct Exists: https://facebook.com/cr00k
[+] Acct Exists: https://www.youtube.com/cr00k
[+] Acct Exists: https://twitter.com/cr00k
[+] Acct Exists: https://www.instagram.com/cr00k
[+] Acct Exists: http://cr00k.blogspot.com/
[+] Acct Exists: https://plus.google.com/+cr00k/posts
```

```
[+] Acct Exists: https://www.reddit.com/user/cr00k/
[+] Acct Exists: https://www.ebay.com/usr/cr00k
[+] Acct Exists: https://cr00k.wordpress.com/
[+] Acct Exists: https://www.pinterest.com/cr00k/
[+] Acct Exists: https://cr00k.yelp.com
[+] Acct Exists: https://github.com/cr00k
[+] Acct Exists: http://cr00k.tumblr.com/
[+] Acct Exists: https://www.producthunt.com/@cr00k
[+] Acct Exists: https://steamcommunity.com/id/cr00k
[+] Acct Exists: https://myspace.com/cr00k
[+] Acct Exists: https://foursquare.com/cr00k
[+] Acct Exists: https://soundcloud.com/cr00k
[+] Acct Exists: https://cash.me/$cr00k/
[+] Acct Exists: https://www.dailymotion.com/cr00k
[+] Acct Exists: https://disqus.com/by/cr00k/
[+] Acct Exists: https://www.deviantart.com/cr00k
[+] Acct Exists: https://www.instructables.com/member/cr00k
[+] Acct Exists: https://keybase.io/cr00k
[+] Acct Exists: https://www.kongregate.com/accounts/cr00k
[+] Acct Exists: https://cr00k.livejournal.com
[+] Acct Exists: https://angel.co/cr00k
[+] Acct Exists: https://www.last.fm/user/cr00k
[+] Acct Exists: https://www.tripit.com/people/cr00k#/profile
[+] Acct Exists: https://fotolog.com/cr00k/
[+] Acct Exists: https://imgur.com/user/cr00k
[X] Could not find required datasets.

[?] Tinder
[+] User: cr00k
[X] No Profile Found.
```

Again, there is a high likelihood that many of these are either the wrong person or simply 404 pages, but each one absolutely needs to be checked out.

One More Username Search

Just for the sake of thoroughness, and also because we don't want him to feel left out, let's also run a username search for CraftyCockney:

```
[?] Whats the target's screenname? : Craftycockney

[?] Knowem
[+] Account: Blogger
[+] Account: Etsy
[+] Account: Hubpages
[+] Account: LinkedIn
[+] Account: MySpace
[+] Account: Photobucket
[+] Account: Pinterest
```



```

[+] Account: reddit
[+] Account: scribd
[+] Account: Tumblr
[+] Account: Twitter
[+] Account: Typepad
[+] Account: Wordpress

[?] NameChk
[+] Acct Exists: https://facebook.com/Craftycockney
[+] Acct Exists: https://twitter.com/Craftycockney
[+] Acct Exists: https://www.instagram.com/Craftycockney
[+] Acct Exists: http://Craftycockney.blogspot.com/
[+] Acct Exists: https://plus.google.com/+Craftycockney/posts
[+] Acct Exists: https://www.reddit.com/user/Craftycockney/
[+] Acct Exists: https://www.ebay.com/usr/Craftycockney
[+] Acct Exists: https://www.pinterest.com/Craftycockney/
[+] Acct Exists: https://Craftycockney.yelp.com
[+] Acct Exists: http://Craftycockney.tumblr.com/
[+] Acct Exists: https://www.producthunt.com/@Craftycockney
[+] Acct Exists: https://myspace.com/Craftycockney
[+] Acct Exists: https://foursquare.com/Craftycockney
[+] Acct Exists: https://www.etsy.com/people/Craftycockney
[+] Acct Exists: https://soundcloud.com/Craftycockney
[+] Acct Exists: https://disqus.com/by/Craftycockney/
[+] Acct Exists: http://photobucket.com/user/Craftycockney/
library
[+] Acct Exists: https://www.deviantart.com/Craftycockney
[+] Acct Exists: https://www.instructables.com/member/
Craftycockney
[+] Acct Exists: https://angel.co/Craftycockney
[+] Acct Exists: https://www.last.fm/user/Craftycockney
[+] Acct Exists: https://www.tripit.com/people/Craftycockney#/
[+] Acct Exists: https://fotolog.com/Craftycockney/
[X] Could not find required datasets.

[?] Tinder
[+] User: Craftycockney
[X] No Profile Found.

```

NOTE When searching for usernames, be sure to always try common letter substitutions. For example, CraftyCockney's Twitter account is actually spelled crafty-cockn3y. This can definitely get tedious, but if you don't try it, you will miss a lot of important accounts. Now that we know Nathan's Twitter account, a quick peek at who he is following immediately tells us that he is into video games—specifically, Assassin's Creed, Xbox, and Ubisoft.

Userrecon

Userrecon is a simple Python script that searches for usernames across 75 different social networks. It is very similar to the Knowem or Namechk tools used by Skiptracer in that the tool will look for active accounts under a number of different sites by testing the URL.

Userrecon tests more sites and has more accurate results, so it is worth trying. When the script is done running, it automatically saves the results to a text file, which is incredibly useful for cataloging or importing to other applications.

You can download Userrecon via GitHub at <https://github.com/thelinuxchoice/userrecon>.

Figure 16.19 shows the initial Userrecon screen with a search term of cr00k.



Figure 16.19

Userrecon returns the following results:

```
[?] Input Username: cr00k
[*] Removing previous file: cr00k.txt
[*] Checking username cr00k on:
[+] Instagram: Found! https://www.instagram.com/cr00k
[+] Facebook: Found! https://www.facebook.com/cr00k
[+] Twitter: Found! https://www.twitter.com/cr00k
[+] YouTube: Found! https://www.youtube.com/cr00k
[+] Blogger: Found! https://cr00k.blogspot.com
[+] GooglePlus: Found! https://plus.google.com/+cr00k/posts
[+] Reddit: Found! https://www.reddit.com/user/cr00k
[+] Wordpress: Found! https://cr00k.wordpress.com
[+] Pinterest: Found! https://www.pinterest.com/cr00k
[+] Github: Found! https://www.github.com/cr00k
[+] Tumblr: Found! https://cr00k.tumblr.com
[+] Flickr: Found! https://www.flickr.com/photos/cr00k
[+] Steam: Found! https://steamcommunity.com/id/cr00k
[+] Vimeo: Not Found!
[+] SoundCloud: Found! https://soundcloud.com/cr00k
[+] Disqus: Found! https://disqus.com/cr00k
[+] Medium: Found! https://medium.com/@cr00k
[+] DeviantART: Found! https://cr00k.deviantart.com
[+] VK: Found! https://vk.com/cr00k
```

```
[+] About.me: Found! https://about.me/cr00k
[+] Imgur: Found! https://imgur.com/user/cr00k
[+] Flipboard: Found! https://flipboard.com/@cr00k
[+] SlideShare: Found! https://slideshare.net/cr00k
[+] Fotolog: Found! https://fotolog.com/cr00k
[+] Spotify: Found! https://open.spotify.com/user/cr00k
[+] MixCloud: Not Found!
[+] Scribd: Not Found!
[+] Badoo: Not Found!
[+] Patreon: Found! https://www.patreon.com/cr00k
[+] BitBucket: Found! https://bitbucket.org/cr00k
[+] DailyMotion: Found! https://www.dailymotion.com/cr00k
[+] Etsy: Found! https://www.etsy.com/shop/cr00k
[+] CashMe: Found! https://cash.me/cr00k
[+] Behance: Not Found!
[+] GoodReads: Not Found!
[+] Instructables: Found! https://www.instructables.com/member/cr00k
[+] Keybase: Found! https://keybase.io/cr00k
[+] Kongregate: Found! https://kongregate.com/accounts/cr00k
[+] LiveJournal: Found! https://cr00k.livejournal.com
[+] AngelList: Found! https://angel.co/cr00k
[+] last.fm: Found! https://last.fm/user/cr00k
[+] Dribbble: Found! https://dribbble.com/cr00k
[+] Codecademy: Found! https://www.codecademy.com/cr00k
[+] Gravatar: Found! https://en.gravatar.com/cr00k
[+] Pastebin: Found! https://pastebin.com/u/cr00k
[+] Foursquare: Not Found!
[+] Roblox: Found! https://foursquare.com/cr00k
[+] Gumroad: Not Found!
[+] Newgrounds: Found! https://cr00k.newgrounds.com
[+] Wattpad: Found! https://www.wattpad.com/user/cr00k
[+] Canva: Found! https://www.canva.com/cr00k
[+] CreativeMarket: Found! https://creativemarket.com/cr00k
[+] Trakt: Found! https://www.trakt.tv/users/cr00k
[+] 500px: Not Found!
[+] Buzzfeed: Found! https://buzzfeed.com/cr00k
[+] TripAdvisor: Found! https://tripadvisor.com/members/cr00k
[+] HubPages: Found! https://cr00k.hubpages.com/
[+] Contently: Not Found!
[+] Houzz: Found! https://houzz.com/user/cr00k
[+] blip.fm: Not Found!
[+] Wikipedia: Found! https://www.wikipedia.org/wiki/User:cr00k
[+] HackerNews: Not Found!
[+] CodeMentor: Not Found!
[+] ReverbNation: Found! https://www.reverbnation.com/cr00k
[+] Designspiration: Not Found!
[+] Bandcamp: Not Found!
[+] ColourLovers: Not Found!
[+] IFTTT: Not Found!
[+] Ebay: Found! https://www.ebay.com/usr/cr00k
```

```
[+] Slack: Not Found!  
[+] OkCupid: Found! https://www.okcupid.com/profile/cr00k  
[+] Trip: Found! https://www.trip.skyscanner.com/user/cr00k  
[+] Ello: Found! https://ello.co/cr00k  
[+] Tracky: Not Found!  
[+] Tripit: Found! https://www.tripit.com/people/cr00k#/profile/  
    basic-info  
[+] Basecamp: Not Found!  
[*] Saved: cr00k.txt
```

Based on the results, it looks like Userrecon searches quite a few more sites than the other tools, and also seems to have a better handle on 404 and invalid pages. This is a great secondary tool to validate results and make sure you have all bases covered.

Reddit Investigator

It never ceases to amaze me how much information I end up finding on Reddit. People post on the craziest topics—you can usually find threat actors on subreddits involving hacking, drugs, and/or video games.

One tool worth keeping in your toolbelt is www.redditinvestigator.com. The site is incredibly simple to use and offers both free and professional versions.

Figure 16.20 shows the Reddit Investigator home page, which contains no more than a single search field.

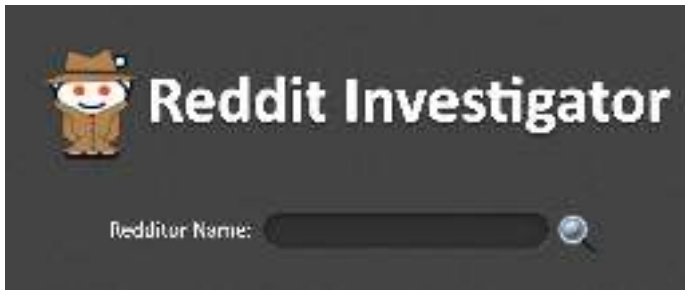


Figure 16.20

If you find yourself using this site often, the professional version is fairly inexpensive and allows you to also search cached posts and view entries over time spans. The ability to search cached (and potentially removed posts) is worth every penny.

Think about it like this: when a threat actor makes a post about some hack or something he is bragging about, then a few weeks later decides to remove it, the value of being able to access that historical data at a later date after it has been removed is enormous.

The output of a search provides statistics on the user, including which subreddits they post on, their karma points, which posts they upvote, most active hours, and much more.

Let's run a search against our good friend WhitePacket. Figures 16.21 and 16.22 show the resulting dashboard stats from WhitePacket's Reddit posts.

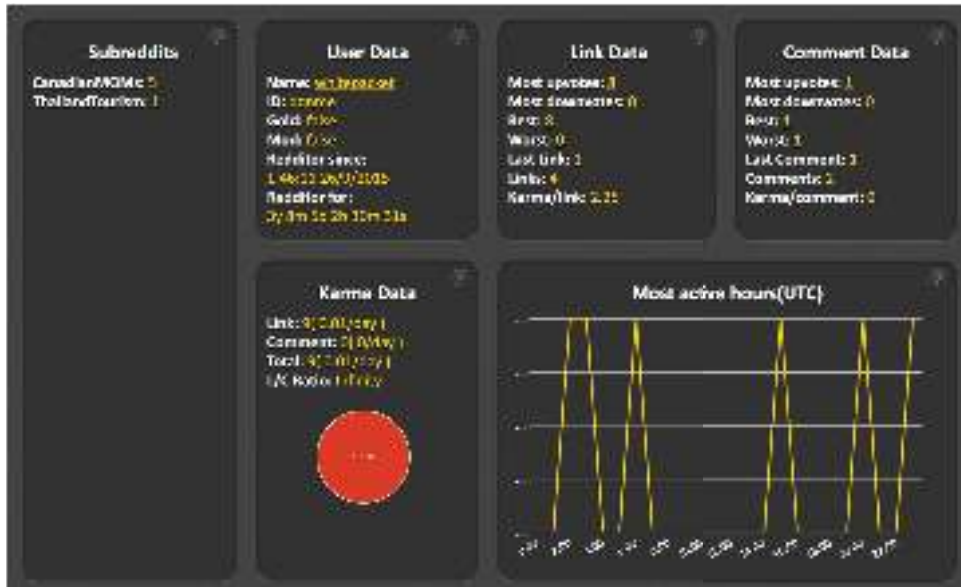


Figure 16.21

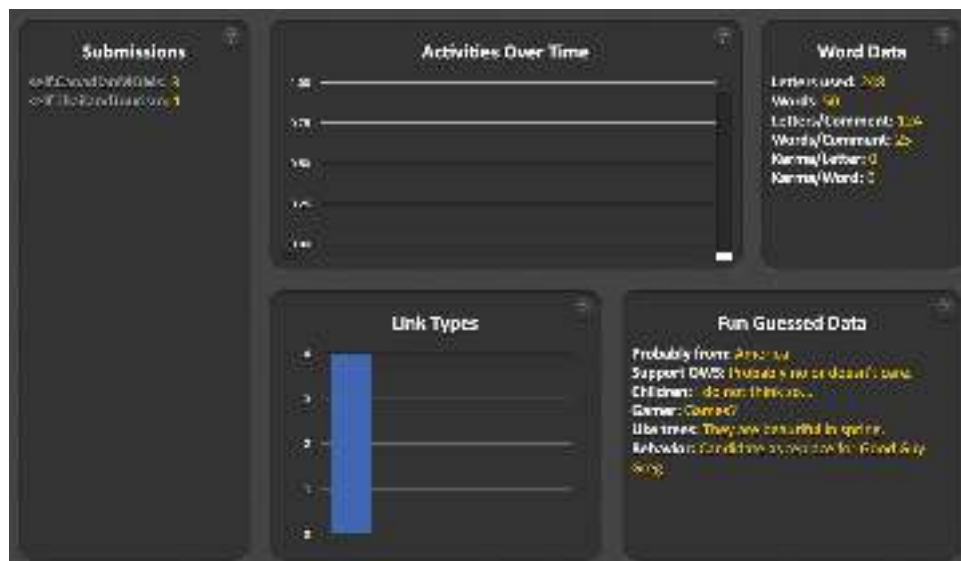


Figure 16.22

Looking at this data, it does not look like the WhitePacket alias is very active on Reddit. It looks like the majority of his posts are on a subreddit called CanadianMOMs. Before you get all excited, this is not a porn subreddit. MOM stands for Mail Order Marijuana.

See that! What did I say earlier about finding people on drug or video game subreddits?

This information is significant because it will allow you not only to track the user's activities and interests but also to see who they interact with. Building a profile on your target's friends and acquaintances can be just as important as building a profile on the actor. One friend leads to two friends, and the more information you have, the more dots you can start to connect.

A Critical "Peace" of the TDO Investigation

To put this into perspective and give a real example of just how useful this type of information can be, the most critical moment of my TDO investigation was when I discovered that user cr00k was actually "Peace of Mind" (aka Peace).

Once I knew to start looking for Peace, I was able to look into who his friends and associates were. It was not until that point that I really understood who was operating within TDO. By this point (somewhere in 2018), there were more than enough threat intel reports suggesting that Peace and Ping (the owner of Hell, a popular darkweb hacker forum) were the same person.

Assuming this information to be true (until I discovered otherwise), I started researching Ping, all of his known associates, and all of the major players from Hell. This led me down the path of needing to find the historical forum data (which included SQL dumps and scrapes). Once I had that, I was able to form a picture of who Ping's close friends were based on the forum conversation.

The old Hell forum posts made it very clear that Ping and another user, Revolt, were very close friends. At one point during the height of some media attention surrounding Hell, Ping made a statement that he was giving full admin control to Revolt while "he was away."

I don't know about you, but I am not giving over admin access to anything unless I really trust the person I am giving it to. Keeping this in mind, if we know that Peace, aka cr00k, is part of TDO, wouldn't it stand to reason that he was also involved with people he was really close with? What better place to start looking than Revolt?

NOTE Full disclosure, this isn't exactly how it happened. I actually came across Revolt's name much earlier in the process, before I even realized the identity of cr00k. It wasn't until I understood the connection between cr00k and Ping that I was able to finally connect the dots. The story isn't false or untrue, I just wrote it that way to help demonstrate why it can be so crucially important to also do your due diligence on your target's friends and known associates.

Summary

This chapter covered the tools and techniques used to leverage social media to help find more information on your target. It would be incredibly rare that a threat actor or target does not have a presence on social media. Chances are they will have more than one account on multiple sites, so it is important to exhaustively check all information you have against as many social sites as possible.

Over the next two chapters, we are going to take all of this information and use it to build a threat actor matrix. Fun!!

Profile Tracking and Password Reset Clues

This is, by far, my favorite part of the OSINT process. The next two chapters will detail my process for building a threat actor profile matrix. What is a threat actor profile matrix, you ask? It is essentially a giant Excel sheet full of the multiple account names, usernames, fake profiles, URLs, and other personal information related to a target. It is a way to visually organize your account-related data into one place. Once you are able to do that, you will start to notice patterns in the data.

We will be using characters and personas from The Dark Overlord group for our examples.

NOTE The remaining chapters will focus on building a profile of a cyber threat actor, which means that the methodology I choose to use will be (mostly) geared toward finding this type of person. If I were looking for a regular person (i.e., non-cyber criminal), I might search different platforms. The important takeaway is the process and techniques.

Where to Start (with TDO)?

You have to start somewhere, right? So if the goal is to investigate TDO (or any threat actor), the first question is “What do we already know?”

One thing we can use to our advantage is all of the press and media attention that TDO received. Looking at early news articles, we know that the original stolen data was being sold on several different hacker forums by a user named Cr00k.

Figure 17.1 is a screenshot of Cr00k's TDO sales thread from the Russian hacking forum Exploit.



Figure 17.1

Similar sales threads exist on several other darkweb hacker forums, including Hell, Siph0n, and 0day. To be clear, when I say similar, I mean the exact same post.

If you find yourself on two different hacker forums and see the same post by two different people, it would not be far-fetched to assume the two users are either the same person or working together.

On two separate darkweb forums, user “F3ttywap” (yes, like the rapper) can be linked to Cr00k through the posting of identical content.

This is where we will begin to assemble our tracking matrix.

Building a Profile Matrix

Depending on the complexity of the person (or people) you are looking for, there can potentially be quite a bit of data to track. I have mentioned the advantages of visually being able to see everything in one place, or on one screen, but the deeper you get into unraveling the mystery of someone’s identity, the less likelihood of that being possible.

Something to remember is that threat actors (especially the seasoned ones) will have many different accounts and personas, which is why it is probably not a bad idea to split up this data across different Excel sheets (tabs). Otherwise the information will become jumbled and you won’t be able to see anything at all.

Keeping that in mind, I typically start with the following three tabs in my matrix:

- Accounts
- Verifications
- Dumps

Each of these three sections will be explained in greater detail over the next two chapters, but at a high level: the Accounts tab is used to identify which email addresses have accounts on specific websites; the Verifications tab keeps track of password reset and verification question information; and the Dumps tab tracks data discovered from password dumps and other hacked data (which is the focus of the next chapter).

Starting a Search with Forums

The first thing we need to do is search. But where? To his credit, the threat actor behind the names Cr00k and F3ttywap was smart enough to use popular names that have plenty of existing search results. Searching Google for F3ttywap will give us a mountain of information on the rapper, which is useless to us. Researching a threat actor under these conditions makes it extremely difficult unless you are a master at dorking techniques.

We need to narrow our search, and given that we know the name F3ttywap from hacker forums, the best place to look for more clues would be other hacker forums.

It is unlikely that you will be able to use dorks and other advanced search engine techniques to find relevant data from forums, because most hacker forums require a login (sometimes paid) in order to view their content. But as with everything else, sometimes you will get lucky. *This is one of those times.*

There are several clear web hacker forums we can (and should) look through, such as HackForums.net, Nulled.to, RaidForums.com, and OGUsers.com. Having some aliases to work with means we can easily perform a Google search using the “insite” operator, or we can just go to each site and search for posts or threads containing that username.

Let’s start with a Google search using Nulled.to as a target site. Figure 17.2 shows us the results of the following search:

```
'f3ttywap' insite:nulled.to
```

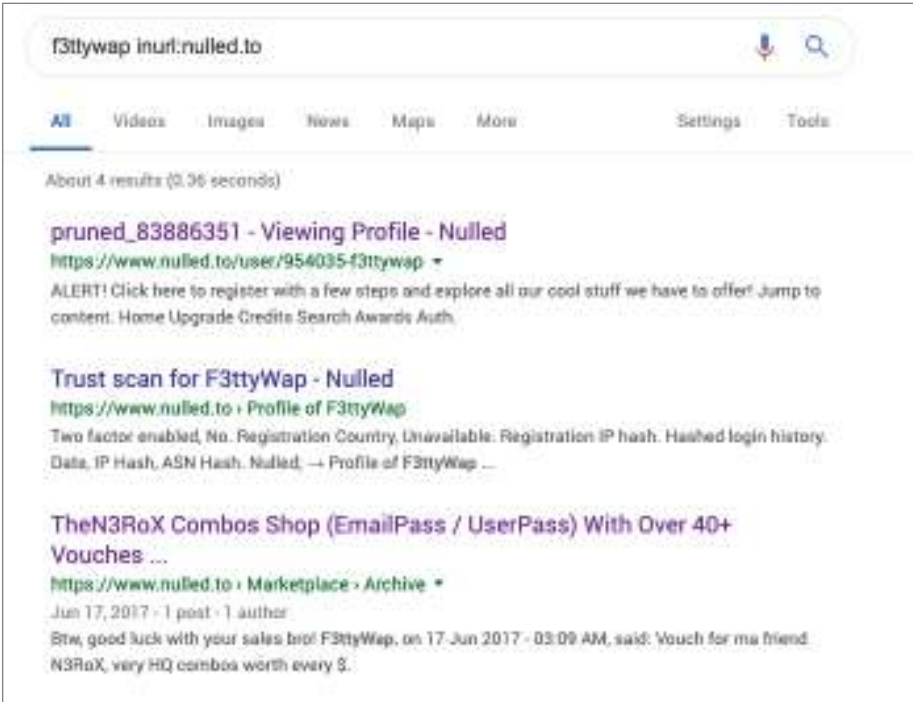


Figure 17.2

The first link takes us right to his profile page: <https://www.nulled.to/user/954035-f3ttywap>. The problem, as we can see, is that the account was pruned (deleted). What we can also see in Figure 17.3, which is a very important detail, is that the user was banned before the account was pruned.

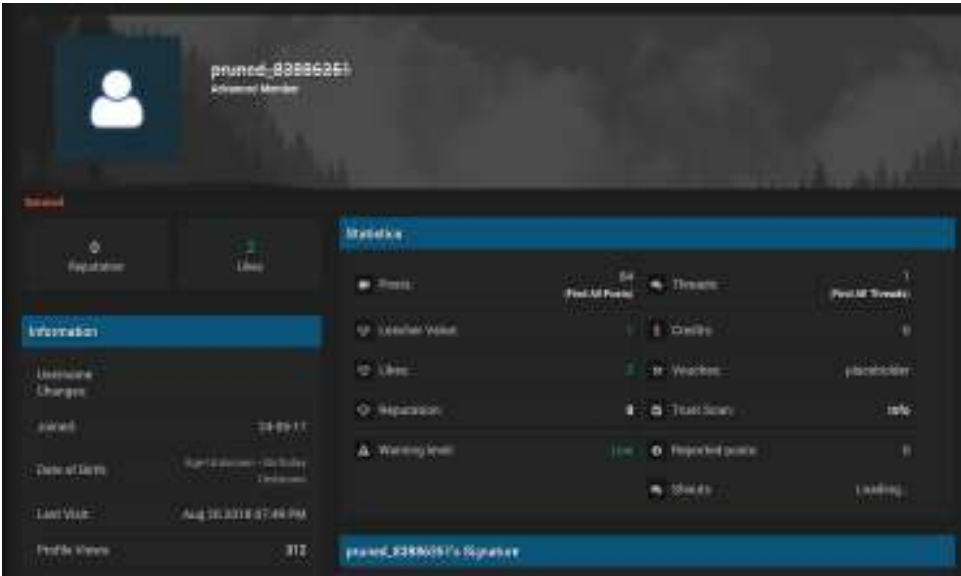
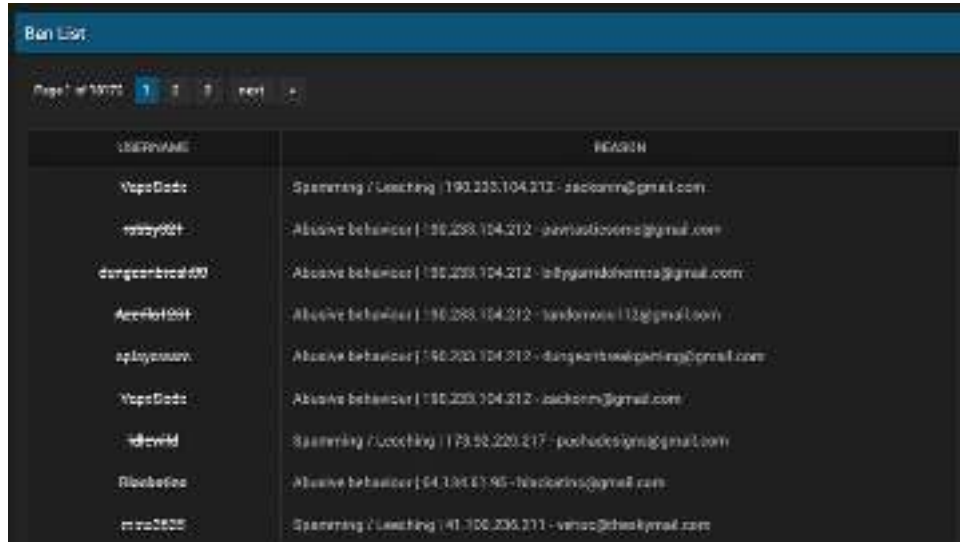


Figure 17.3

Ban Lists

Most hacker forums have a *ban list*, which is an aggregate list of users that have been banned from the site. As a deterrent to stop scammers from trolling their sites, most forums will publish a banned user's registered email address alongside the banned username (ironic, I know).

Figure 17.4 shows Nulled.to's ban list, which can be accessed using the "ban list" link in the site's footer.



USERNAME	REASON
VapeDada	Spamming / Leeching 193.235.104.212 - sadconna@gmail.com
reayy00t	Abusive behavior 193.235.104.212 - anastasiavere@gmail.com
dangearth000	Abusive behavior 193.235.104.212 - bitygandoferrera@gmail.com
Acce0t00t	Abusive behavior 193.235.104.212 - random00112@gmail.com
spinyman	Abusive behavior 193.235.104.212 - dangearth000@gmail.com
VapeDada	Abusive behavior 193.235.104.212 - sadconna@gmail.com
idewild	Spamming / Leeching 179.52.220.217 - pushdesigns@gmail.com
Blacketoe	Abusive behavior 64.134.43.95 - blacketoe@gmail.com
mr0000	Spamming / Leeching 41.106.235.211 - vsmc@theolymp.com

Figure 17.4

Since we know that f3ttywap was banned, we can now reasonably expect to find his email address by looking through this ban list. The immediate problem is that the ban list is more than 18,000 pages long, and there is no search function. To solve this problem, another approach was needed: one that involved getting the information direct from one of the site's moderators.

Social Engineering

Being able to socially engineer your target (or their associates) in order to extract a vital piece of information will more than likely be a necessity. It certainly was in this case. Before we get into the details of my story, let's kick off this section properly with a word from the social engineering master, Mr. Human Hacker, Chris Hadnagy.

EXPERT TIP: CHRIS HADNAGY

We recently did an attack on a company where our goal was to see if we could reset a password. To reset the password, we needed an employee ID plus the answer to one of their security questions.

The way that we worked it was we first looked for an employee that had their employee ID out there. Either a picture of their badge or they just posted it somewhere, because it was a mistake in an email or an email thread or something to that effect. We found a few.

Then, we looked at the people we found and determined if they had a lot of social media presence. I can't remember the exact number, but let's say we found 10 people, and then 5 of those people had a big presence on social media.

Then, we doxed those five, and we called the password change line and made believe we were one of those people. The conversation was basically like, "Hey, this is Vinny. I'm on the road right now. I can't do it, because I'm not at my computer, but I have to reset my password, because I forgot it. What am I going to need again later?"

Every time, no doubt, the person on the end of the phone told us, "You'll need your employee ID plus the answer to one of your security questions." Then we just had to say, "Oh, man, I can never remember what I put for those answers. What did I do?"

Then they usually tell us the answers to one or two of those security questions.

No kidding, every time.

Then, we go and look through our information that we doxed, see if we got that tidbit. If not, find it, then call back later, and when we call back, we have the answer.

We just did this attack last week, and it worked flawlessly. We were able to reset passwords because of that, making our test a success.

In the case of F3ttywap, I was stuck and needed the email address associated with his username. So I did what any respectable hacker would do, and logged in to the site's support Discord page. I wish I'd saved a copy of the conversation, but I didn't.

Long story short, one of the site's moderators was willing to look up the account for me. I think I convinced him that a similar user ripped me off and I wanted to make sure it was the same person. He got back to me with a screenshot of the account's database table, showing an email address of kunt.x7@gmail.com. The admin was also nice enough to give me the IP address that was used to register the account!

I guess sometimes, all you have to do is ask!

Now that we have more information to associate with our user, we can start filling out our threat actor matrix. Figure 17.5 shows what my Excel table looks like with this information.

	A	B	C	D
1	Website	Email	Username	Notes
2	kunt.x7	kunt.x7@gmail.com	F3ttywap	95.85.196.212

Figure 17.5

This page of the matrix does not need to be anything overly complex. Right now all we are doing is documenting the data we find. We will beef up the fields in the coming pages.

SE'ing Threat Actors: The “Argon” Story

While we are on the subject of social engineering, I wanted to circle back to our good pal Cyper, and how I was able to infiltrate the KickAss forum and their inner circle.

Around October 2018, a security reporter wrote a story about me titled “When Security Researchers Pose as Cybercrooks, Who Can Tell the Difference?”

At the time, I was using the alias “SoundCard” on the KickAss forum and posing as a data broker. The author of the article exposed my alias, and subsequently put the KickAss members on high alert.

NOTE The article alluded to me posing as a data broker and offering to sell a data leak involving LinkedIn data, all of which was true. What really annoyed me about the article is how quick the author was to throw me under the bus for “throwing his name around,” stating that I somehow “forced his hand in writing the article.”

I don’t even understand what that means, but I am choosing not to say his name because that will only reinforce his belief that I willingly throw his name around, like some prized possession.

The reality is that threat intel professionals pose as cyber crooks all the time in order to leverage their trust in underground communities to gain valuable intelligence.

It is fairly common practice, and this reporter had no problem taking any of my information in the past, or the information from countless other threat intelligence firms and people he receives on a daily basis. Somehow my actions were different?

I asked him what was really up with the article. He emailed me claiming that a close business associate told him that I told them that he was “co-authoring” this book, and that somehow sent him over the edge.

That never happened. Given the context of his statement, I figured out who the associate was, and I can see how the situation could have been easily confused. I did tell this person that the reporter had agreed to provide a guest expert comment on my book, as did several other people. It was more of a “the following people are providing guest contributions” statement.

Either way, I forwarded this reporter the email where he *accepted my invitation* to provide a comment on this book, and that was the last time I ever heard from him.

This was not the first or the last time this author published someone’s personal information, perhaps because he simply did not agree with their point of view. In a recent tweet, this reporter also published the personal information of security researcher NotDan, causing an out cry within the security community.

The point of this rant is: be careful how much information you share and with who. There can be significant consequences in publishing someone's personal information. The result of this particular article, as we will see, set off a chain of events intended to causing me "pain."

The article sparked a lot of interest on the KickAss forum and personally made me a target for attacks. When all of this went down, Cyper (aka NSA), the forum's admin, decided to put out a bounty on my site with the forum post "Own this rat site" (Figure 17.6).

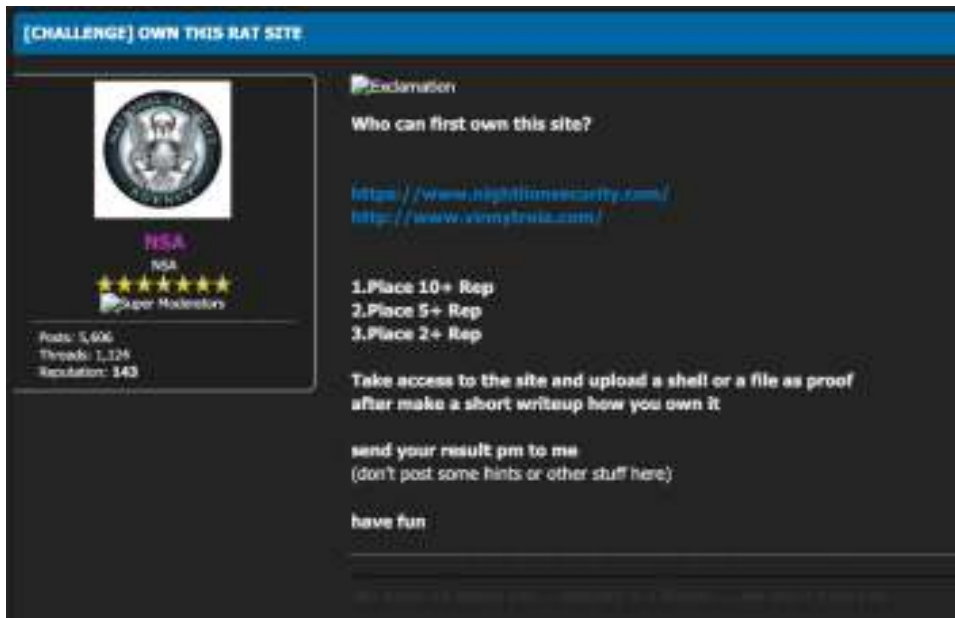


Figure 17.6

To my point regarding the publishing of the article, I can't say I appreciate the fact that I was personally exposed. There are consequences to publicly exposing someone's personal information, as evidenced by the following post by user "deafrow" (Figure 17.7).

Luckily, I am one of those people who believe there is an opportunity in every problem—and in this case, I was right.

Unbeknownst to all, I had a second alias on the forum, "Argon." At this time, Argon was undergoing an application process to be accepted into Cyper's personal hacking group. The contest against my site proved to be a amazing opportunity. What better way to flex my skills as a blackhat ninja hacker than to *hack my own site*?

So I did.



Figure 17.7

I set up an outdated PHPMyAdmin panel in a remote directory on one of my sites. Anyone performing server recon and directory bruteforcing using Wfuzz (as described in Chapter 7) would have been able to find it.

My cover story was simple. Once I found the panel, I ran a wide search for any email/password combinations that could be discovered from my email addresses (this technique will be discussed in the next chapter).

I set the PHPMyAdmin login to an older username/password combo from a random data leak. Using the passwords “discovered” from various data leaks, I pretended to gain access by bruteforcing the PHPMyAdmin panel.

To prove I was there, I “hacked” my own WordPress site by directly modifying the WP database table, and replaced the front page with the image shown in Figure 17.8.



Figure 17.8

But why stop there? I assumed Cyper and his team would want to personally verify the access and login themselves, so prior to my victory announcement I took the liberty of setting up strict firewall rules on my server designed to completely block any connections from known TOR, VPN IP, proxy, or other known bad IPs. I then set up logging to monitor for any incoming connections.

When the trap was set, I made a celebratory post on the forum's contest thread (Figure 17.9).



Figure 17.9

Not bad, right? How many people would expect someone to hack their own website?

NSA asked me a lot of questions about how I was able to do it, all of which was completely plausible; and it worked!

NSA and his team had their fun logging in and playing around my panel, and as my reward, I got to record all of their IP information. The PHPMyAdmin URL was so obscure (and so brand new) that any hits were obviously relevant—which included the fake web scraping bots that suddenly appeared.

The point of this story is that if you are going to SE a target, why hold back? *Sometimes the most outrageous ideas are the best ones.*

EXPERT TIP: JOHN STRAND

When you start talking about op-sec for adversaries, it isn't just a matter of op-sec from the perspective of getting them to open any old document and using that to hopefully gather some information about them. It's a matter of using the right bait. I've learned over the years of doing this (for quite a while) that *fake documents look fake*. If you can actually put tracking elements inside a real document, that's a lot better.

Macros will freak an adversary out. If you try to drop a macro in a document, they're not going to open it. If you try to put a macro in an Excel spreadsheet, well, macros are always in Excel spreadsheets . . . so are they going to open it, or are they not? It's a different scenario.

On the other hand, if you tried to put in a link to a cascading style sheet (CSS), or an image source tag in the metadata of a document, your likelihood of them actually opening that document and having it beacon back is now a lot better.

Some of my favorite ideas for trying to get adversaries to run things have come down to really simple ideas. Just set up something like a directory on a website that says VPN, and have an executable that does the automatic VPN configuration and then you create that executable with its own callback functionality. Or you can even digitally sign that executable, then wait for any lookups on that digital code signing certificate whenever it fires, or whenever somebody tries to run the executable as well.

It's more of an issue of just moving beyond the traditional document rouse, but actually using the right bait to try to get that individual to open up those particular documents and then interacting with a component that you created just to attract them.

Everyone Gets SE'd—a Lesson Learned

If you are paying attention to the story, at some point you may ask yourself, "Wait a second. The article exposed your handle as "SoundCard" but made no mention of Argon. How did that name get out?" Well, funny enough, I was SE'd.

In the midst of all the excitement, Cyper decided to jump on Twitter and chime in on the discussion between Krebs and me. Figure 17.10 shows a copy of Cyper's message from the KickAss_Sec Twitter page (which has since been closed).

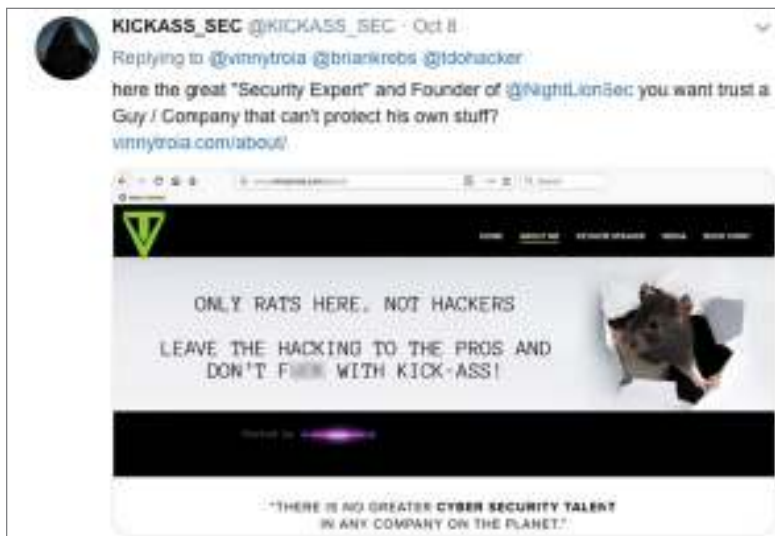


Figure 17.10

Somewhere in that heated Twitter conversation, Cyper dropped a promo for the KickAss forum (but used the old URL, which I immediately recognized). I clicked the link to see what was up and why he would be using the old address. My TOR browser had cookies saved, and I was immediately logged into the old KickAss URL as Argon.

Well played, sir.

The moral of the story—don't store cookies in TOR, and *don't be so quick to click on links!* (Duh, right?)

The End of TDO and the KickAss Forum

In my official report on TDO, I propose a theory that the KickAss forum, which was the official home to The Dark Overlord, eventually quit due to an elaborate exit scam.

NOTE An *exit scam* (at least in this context) involves the owner of a darkweb marketplace suddenly closing down the site and keeping all of the money in its escrow account.

Darkweb marketplaces sell illegal merchandise (usually drugs) and run on an escrow system. The buyer pays for their items with cryptocurrency that is then stored in the site's escrow account. When the buyer receives the items, they acknowledge receipt of the goods and release the funds to the seller.

In an exit scam, once a marketplace has enough transactions and money stored its escrow account, the admin will shut down the site, keeping all of the money and screwing the vendors.

I believe this is why the KickAss forum ultimately closed.

TDO's final media campaign involved leaking layers of insurance information from the 9/11 attack, dubbed as "stolen 9/11 papers." Each layer, when released, contained a different set of *worthless* insurance documents. The group used this tactic to lure the media hoping to gain attention by preying on 9/11 conspiracy theorists.

It worked . . . for a short time.

Each layer of documents contained a PGP key to verify that the data was legitimately coming from the TDO, and also contained the following official contact information:

```
CONTACT AND LOCATION DETAILS:
thedarkoverlord E-Mail Address: tdohackers@protonmail.com
Backup1 E-Mail Address: thedarkoverlord@msgsafe.io
Backup2 E-Mail Address: thedarkoverlord@torbox3uiot6wchz.onion
Make your own at (torbox3uiot6wchz.onion)
KickAss Tor Address: kickassugvgoftuk.onion
```

How interesting that TDO would suddenly be promoting the KickAss forum. Even more interesting was the fact that they were promoting the *old* URL (the same one that I accidentally clicked on earlier in this chapter).

But all things must eventually come to an end, and after about a month, the site apparently closed for good . . . or maybe it didn't?

Funny enough, a few days ago, a friend of mine was having a conversation with a random threat actor and my name came up. The following is a transcript of their conversation.

```
X: u know if kickass is back yet?
X: i cant find NSA or inferno anywhere
XXX: Imo, kickass isn't coming back
X: oh
X: they start a new site?
X: or all the TDO s***?
X: it looked like they were exit scamming
XXX: With all the s*** that went down with krebs/troy everythings on
    pause
XXX: I think it was tdo related
X: what went down with krebs / troy?
XXX: Thats what I heard anyway
XXX: well
XXX: less them, more that guy Vinny troya
XXX: troia*
X: oh the article
X: gotcha
```

Oops!

I blame Krebs, anyway since it was his article.

Now let's get back to our regularly scheduled chapter content on using password reset clues to fill in our tracking matrix.

Using Password Reset Clues

What's a password reset clue? Whenever you click the "forgot password" option on a website, the site usually provides you with a clue followed by a prompt to verify your identity. Another term for these clues might be "account verification questions."

For example, the website in question might say something like "in order to reset your password, we are going to send an email to your email address v*****@gmail.com."

But what if you don't have access to that email anymore? It is a completely plausible scenario, which is why there are typically two or three different verification options, which can include emails, text messages, and answering security questions.

These clues can provide an enormous amount of information when you are building a profile on an individual—and sometimes (as we will see), learning

that no account exists can be equally important in understanding a target's movements.

Starting Your Verification Sheet

Earlier in this chapter we started building a tracking matrix. One of the sheets in the Excel document is labeled Verifications. This Excel sheet is all about collecting and documenting password reset clues.

In the left column of my Verifications tab, I include a list of sites that I typically check for clues. These may vary depending on your target, but a good starter list includes the following:

- Gmail
- Yahoo
- Hotmail/Microsoft
- eBay
- PayPal
- Facebook
- Instagram
- Yahoo
- Twitter
- VK
- Venmo
- LinkedIn
- Steam
- PlayStation Network
- ICQ

Let's start by applying this to our search for F3ttywap. Since we now have an email address to go on (kunt.x7@gmail.com), a good place to start the Verifications sheet of our tracking matrix is with Google's Gmail.

Gmail

Starting with Gmail, we can enter in our target's email address and click the "forgot password" button to get to the account recovery page. Figure 17.13 shows the first Gmail account recovery screen.

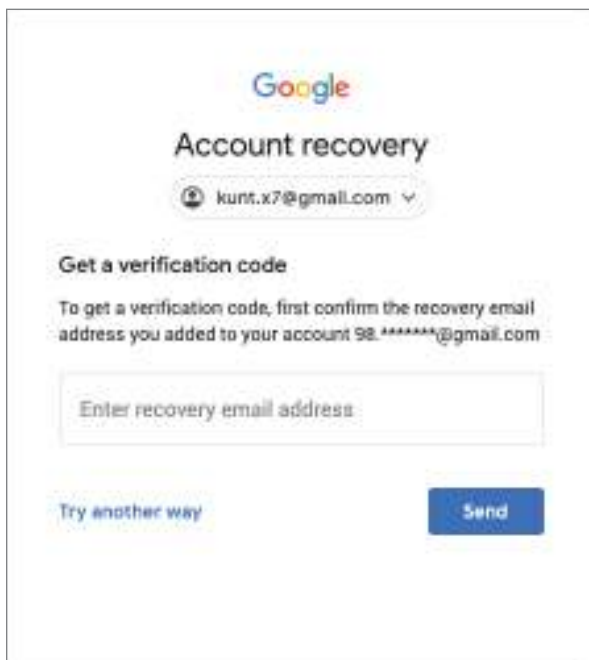


Figure 17.13

We now have our first clue. The recovery email is 98.*****@gmail.com.

NOTE The great thing about Gmail (and most account recovery sites) is that it provides an exact number of characters in the recovery clue. This will allow us to connect certain dots later in our investigation without having to worry about making unnecessary assumptions.

Most sites provide more than one method for account recovery. When working with Gmail, clicking “Try another way” will show additional account validation options.

In this case our test account only had one form of account verification available, but to show you an additional example, Figure 17.14 uses my personal Gmail account to show a second verification option.

We can see from this screen that my phone number is 12 digits (important to note when dealing with international numbers), and ends in 68.

Now we have enough information to start filling in the Verifications tab on our tracking matrix. Figure 17.15 shows what I have added to my tracking matrix.

I included my own phone number in the verification number column as an example of how I would fill in this data.

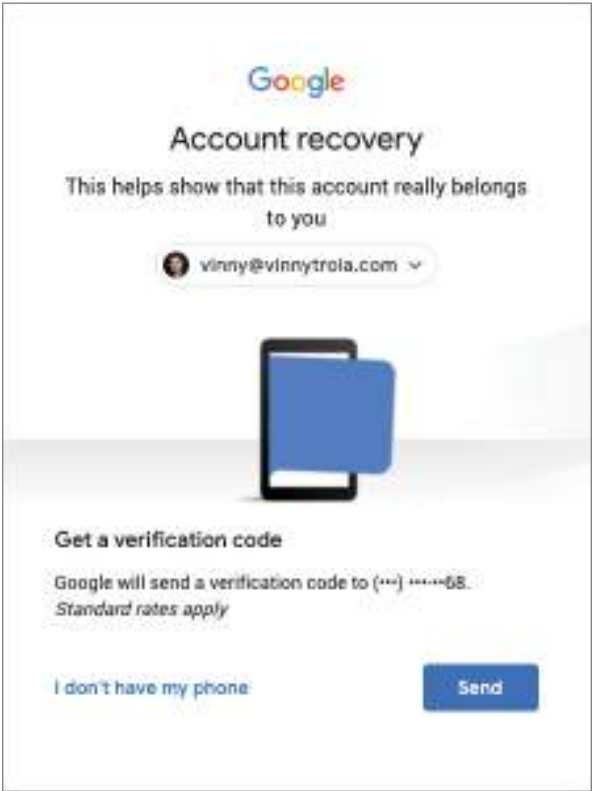


Figure 17.14

Site	Email	Username	Verification email	Verification Number
Gmail	kunt.x7@gmail.com		98.*****@gmail.com	*****68

Figure 17.15

Facebook

Facebook isn't the most helpful when it comes to account verification clues. However, we can still learn a lot even when there is no information to be displayed—a lack of information can be just as significant. Recently, Facebook removed the ability to search for a user by an email address, which means we have no real way of knowing whether a person has a Facebook account using a given email address.

In this case, Facebook's password recovery page will help us accomplish the same task. Figure 17.16 shows the Facebook account recovery page with our target email address entered.

A screenshot of a web form titled "Find Your Account". Below the title is a text input field containing the email address "kunt.a7@gmail.com". To the right of the input field are two buttons: "Search" and "Cancel". Above the input field, there is a line of text that reads "Please enter your email or phone number to search for your account."**Figure 17.16**A screenshot of the same "Find Your Account" web form. A red rectangular box is overlaid on the form, containing the text "No Search Results" and "Your search did not return any results. Please try again with other information." Below this box, the text input field now contains the placeholder text "Phone or email". The "Search" and "Cancel" buttons remain at the bottom right.**Figure 17.17**

If the account exists, we would see a message saying that an email was sent for verification. Instead, we see the message shown in Figure 17.17.

This tells us definitively that this email address does not have an account with Facebook. If the email address did exist, the next step would be to use either Facebook's internal search or an external tool like OSINT.rest or PIPL.com to try to locate the person by searching for the email address.

PayPal

Another very important site to check is PayPal. Most (if not all) threat actors have multiple PayPal accounts, so there is a good chance you will be able to gain important clues using their account recovery options.

Figure 17.18 shows the first PayPal account security check we receive after entering our target's email address in PayPal's "forgot username/password" page.

This is incredibly important because PayPal gave us the last two letters of the first part of the recovery email address, rather than the first three given to us by Gmail. The number of asterisks is the same, which is a good indication that PayPal also shows us the exact number of characters in the recovery email.



Figure 17.18

It is certainly possible that the two sites have different recovery addresses, but assuming for a moment they are the same, we can now extrapolate that the recovery address is 98.****op@gmail.com.

This is a really good lead, one that we will end up uncovering in the next chapter. For now, let's keep going and see what else we can find.

Since we don't have access to the email address, we can select the "Answer your security questions" option. Figure 17.19 shows this prompt.



Figure 17.19

We should document these two clue questions in our matrix. It's possible we might come across some information later that can be used to fill in the details.

Our test email address does not have much associated with its PayPal account, or we would be able to learn more from these clues. To show you what I mean, let's switch back to using my personal account.

Entering my personal email address into PayPal, I am presented with four different ways to verify my identity. The first option is for text message verification. Figure 17.20 shows the PayPal screen, which displays five digits of my phone number.

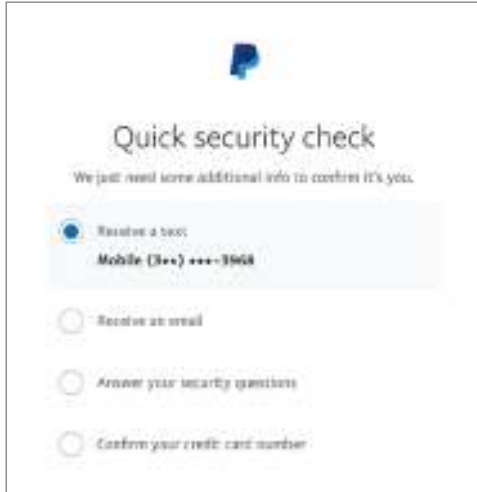


Figure 17.20

Comparing this with what we learned from Gmail, we can identify that the phone numbers both end in 68. This is a pretty good indication that both phone numbers are the same. Now we have the full last four digits of the number, *and* the first digit of the area code. This information in itself can be used to verify an identity on other sites. How many times have you been asked to enter the last four digits of your phone number?

The next option is an email verification check. This is similar to verification options on other sites—you will be shown a portion of the email address and asked to receive an email (shown in Figure 17.21).

Looking at the details provided in Figure 17.21 we can safely assume the recovery domain is mac.com, and we know that the last two letters of the email address are “ia.” Since there are eight asterisks (for a total of nine letters) it would not take much to guess that my recovery email is my name, vinny-troia@mac.com.

The last option in Figure 17.22 is to verify the credit card number on file.

Most sites do not offer this for a verification option, but in case it ever comes up, we now know that I have a Visa that ends in 10.

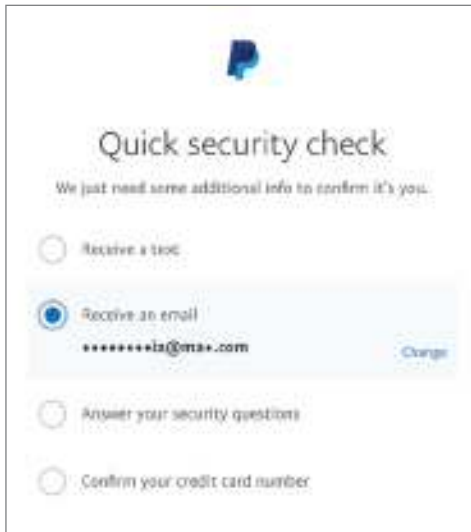


Figure 17.21

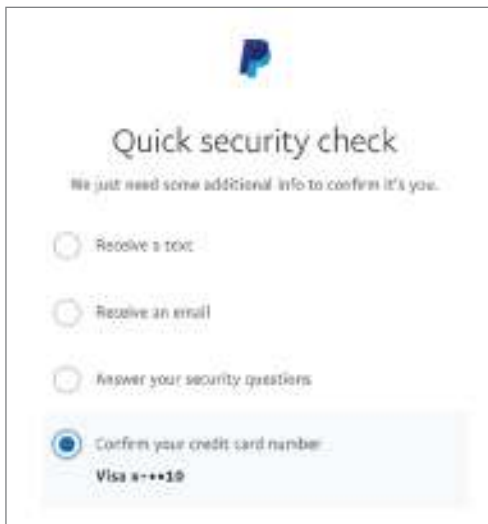


Figure 17.22

Twitter

It is almost a given that threat actors will have multiple Twitter accounts. We can try to find a Twitter account using Twitter's normal search engine, but there is a good chance the name will be obfuscated in some way, or the person will be using an alias that you have not previously encountered.

To give you a great example, at one time I was searching for an old member of the Hell forum known as Revolt. I ran a number of Twitter searches to find his account, but always came up short.

One day, I tried searching for one of his known associates, “Ping,” or more specifically, “Pinger.” I got lucky and found Revolt’s Twitter page by reading through the different posts. His Twitter username was “RevoIt_.” Looking at his page in Figure 17.23, you can see that a capital I looks identical to a lowercase L on Twitter.



Figure 17.23

Back to our search, when using the Twitter’s account reset tools, we are told whether an account exists. Figure 17.24 shows the screen we receive when searching for our target email of kunt.x7@gmail.com.

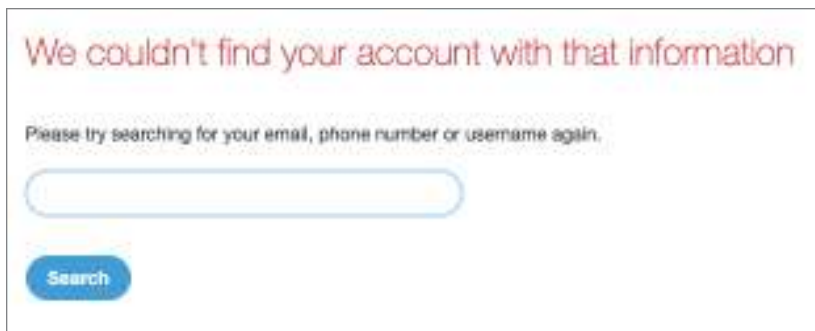


Figure 17.24

But in the case of Twitter, searching for a username can be just as effective. As an example, let's say we want to see if an account exists for someone with the username m4l1h4ck3r. Figure 17.25 shows the response we receive from Twitter.

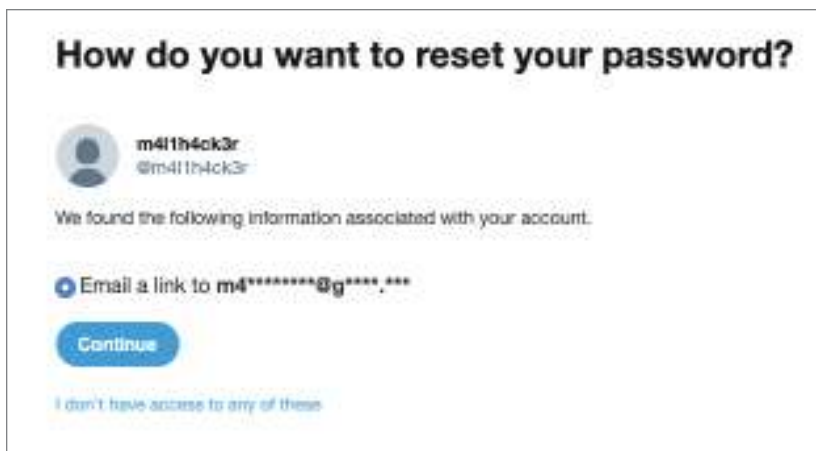


Figure 17.25

Now we have the start of a recovery email address for this username that we can add to our verification matrix.

Microsoft

Another almost guaranteed hit is looking for Hotmail or Skype accounts. Skype accounts were heavily used by threat actors a few years ago before Microsoft did away with Skype's group chat (and pretty much destroyed the product). Even though most threat actors have moved away from Skype to Telegram or Discord, the accounts are most likely still available.

A Microsoft account can be searched by username, email address, or phone number. Figure 17.26 shows the identity verification screen of one of my personal Microsoft accounts.

Unlike other sites, Microsoft discloses the full domain name. If the target was using something like gmail.com or yahoo.com, it would not be a big deal. But in the case of using one of my personal accounts, the verification domain is not something that would be easily guessed.

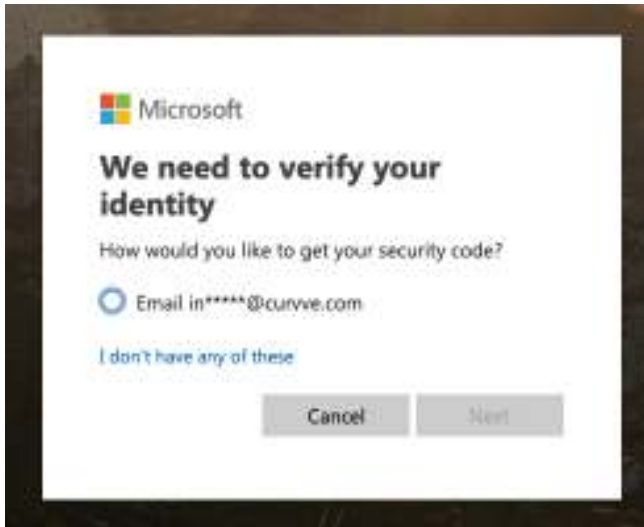


Figure 17.26

Instagram

Much like Facebook or Twitter, most people have Instagram accounts. Even though Instagram won't provide you any clues to enhance your target's profile, the account reset option will at least tell you whether the account exists. To demonstrate, Figure 17.27 shows what happens when I enter a nonexistent email address in the "forgot password" screen.

Looking at the bottom of the page, we can see that no users were found with this email address.

Next, let's try entering a real target email address. Figure 17.28 shows the difference in response.

We can see that this email address exists in Instagram. Even though we can't search for users directly via email address, we at least know the user has an account and can look for other creative ways to find it.

Using jQuery Website Responses

jQuery is a modern JavaScript library. At a very high level, jQuery is designed to simplify JavaScript coding for websites. One common use of jQuery is to provide real-time responses to web form questions, without ever having to submit the form.

For example, have you ever been on a website that asks you to select a username or email address, and as soon as you type in your response, you see a red notice that says "Sorry, this username is already taken. Please try again"?

I find this modern web convenience to be an incredibly useful way to *quietly* determine whether a user has an account on a particular website.

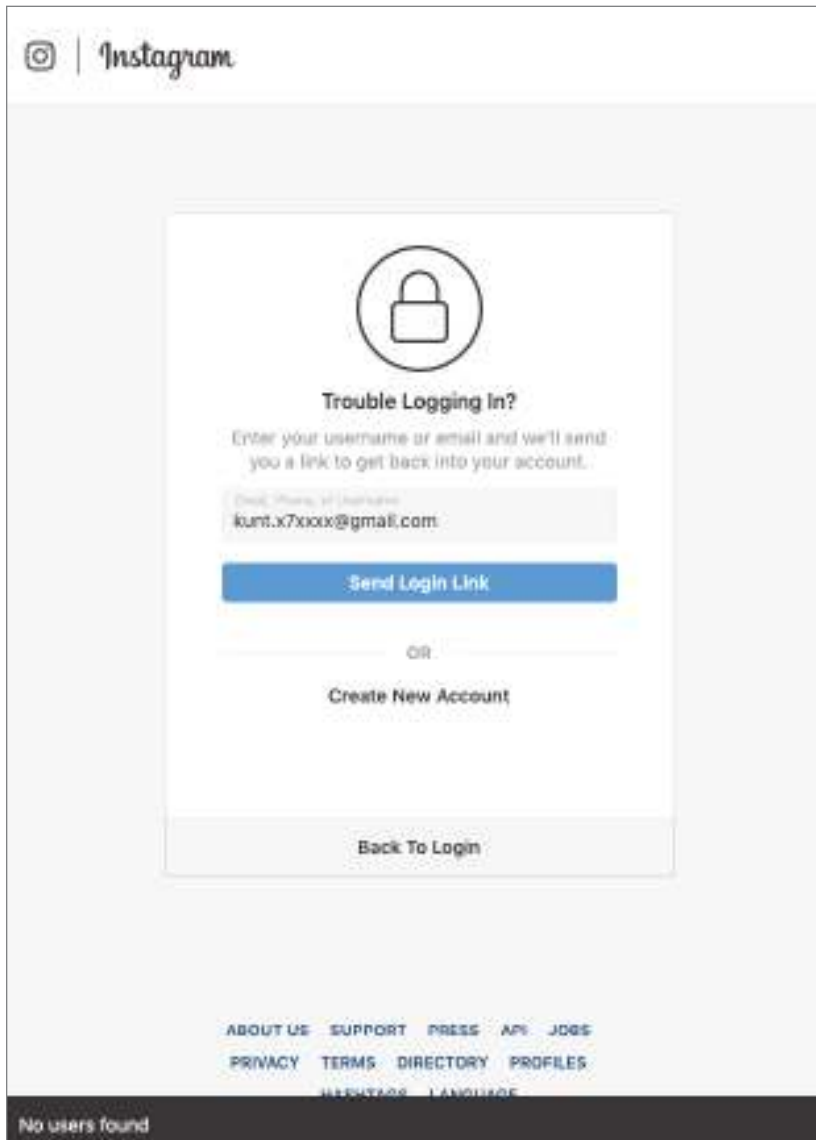


Figure 17.27

To demonstrate, let's see if our test email address is registered on Exploit.in, a popular Russian hacking forum.

Figure 17.29 shows a part of Exploit's signup screen.

Thanks to the marvel of jQuery, we can receive instant verification on whether a username or email address already exists on the forum, without the need of being logged in (and potentially alerting the admins of our searches). Figure 17.30 shows the results of entering our test subject's email address and potential username into the appropriate fields.

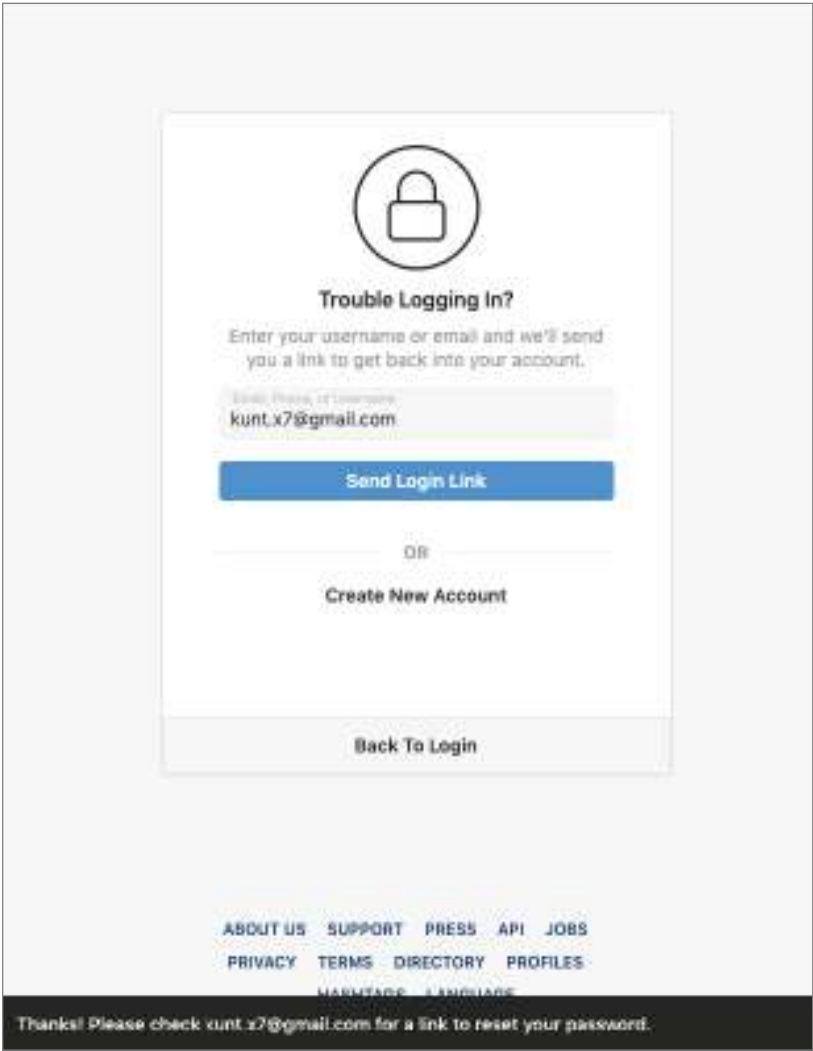


Figure 17.28



Figure 17.29



Display Name * Kunt,x7

Email Address * kunt,x7@gmail.com

Figure 17.30

Both fields came back as green, meaning they are not in use and are available to be registered. Now let's see what happens if we want to see if there is a user with the name "cr00k" on the forum (Figure 17.31).



Sign Up

Display Name * cr00k

⛔ That display name is in use by another member.

Email Address *

Figure 17.31

We can immediately see that the name cr00k is in use by another member of the forum.

These types of instant name checks are available on a large majority of sites on the Internet, and can be used to help determine whether a particular website is of interest in your investigation.

NOTE I typically store this information in the Accounts section of the tracking matrix, but I chose to include this topic here (rather than at the beginning of the chapter) because we had not discovered the account email yet.

ICQ

Last but not least is ICQ, an older chat service from the late 1990s that seems to have survived and is still used heavily by international threat actors. ICQ usernames consist of random 8-to-10-digit numbers, so they are impossible to guess. But sometimes, the less intelligent threat actors will do the work for us.

In Figure 17.32, threat actor cr00k (not the same cr00k that is affiliated with TDO) posted a message advertising the sale of Canadian and U.S. credit cards. In the thread, he posts his ICQ number as a way to contact him.



Figure 17.32

ICQ's account recovery page will allow you to use the account number as a means to initiate the request. Figure 17.33 shows the resulting message when we enter 610-476-385 into ICQ's account recovery page.



Figure 17.33

We can see the first letter of the user's email address, and the full domain name. If we already knew the threat actor's email address, this would be a perfect way to tie him to the ICQ number.

NOTE With respect to this particular ICQ account, it links back to Zain M****, another carder associated with the name cr00k. The email address is a match to his personal live address, zainm****@live.com.

Summary

This chapter discussed the building of a tracking matrix, which should be used to keep all pertinent information on your target. The Excel-based tracking matrix contains three distinct sections: Accounts, Verifications, and Dumps.

This chapter covered several techniques used to help fill up your Accounts and Verifications sections. The Accounts section is used as an inventory of all the sites where your target has an account, while the Verifications section is used to catalog the “account reset” clues provided by those sites.

The next chapter will focus on using the data leaked from hacked websites (aka password dumps) to help us complete our threat actor profile.

Passwords, Dumps, and Data Viper

I find it to be so amazingly cool to be able to say the following sentence: This is the chapter where we get to use a hacker's own hacks against them. What I mean by that is we will be using the data within a "data dump" (i.e., a company's hacked data) to fill in the missing pieces from the previous chapter.

During my quest to uncover the identities of The Dark Overlord, I could not find a single tool with all of the historical data needed to form the necessary conclusions, so I built my own: Data Viper (www.dataviper.io).

Though this tool is not commercially available (yet), I will discuss how I built the tool, how you can build your own, and of course, how it was used to reach conclusions during my investigation.

The following is an excerpt from my first conversation with TDO, circa November 2017, where he and I discuss Cr00k and his affiliation with the group.

TDO: Peace, now that's a fascinating thought.

TDO: We've contracted a great deal of individuals to front for us as data brokers. It's difficult finding the time to do these things when we're busy climbing out way up the hacking chain.

VT: oh, that's interesting. i guess i never looked at it that way.

VT: ok, so if he is a broker, why is he no longer being used?

TDO: As a business owner, we're surprised you hadn't considered the 'supply chain' methodology and its benefits.

VT: i guess i had no idea how organized the group is.

TDO: Do you believe it's an intelligent decision to allow others to adsorb the disadvantages and costs of stylometry, metadata collection, and other sorts of HUMINT?

VT: absorb them how? i am not following?

TDO: Risk.

V: sure, it makes complete sense from the standpoint of anonymity

V: but that kind of brings me back to my question. if we are going to discuss a previous broker, it is important to know why they were selected and why they are no longer involved

TDO: We believe our brokers sub-leased much of the work, even. It's difficult for us to acquire an accurate picture of everything.

VT: so are you still working with cr00k, in any capacity?

TDO: We're attempting to perform a CBA at the moment, and we're having a difficult time understanding any benefits of us answering your question about cr00k.

TDO: Peace, though, how did you come to that one?

TDO: That would make us privy to the likes of some of the planet's largest breaches.

VT: it seems like you guys certainly have the skill.

TDO: Is this something your wife told her school's sports team?

VT: my wife doesn't go to school?

VT: oh. ha ha.

TDO: It's a bit of jest, mate. We're cracking one on you. We assume she did go to school previously.

The really interesting part of this conversation is how TDO zoned in about me asking him about "Peace of Mind," and his eagerness to classify Peace as one of the greatest hackers of all time. I always wondered what that was about.

Now, with the information available in Data Viper, this chapter will answer that question.

Using Passwords

One inevitability known by the people within the security industry is that *everyone*, whether everyday end users, experienced technical users, threat actors, or anyone in between, *will*, at some point, reuse passwords.

As a labeled security "expert" (I say that in quotes because I think the term itself is incredibly pompous), I advise everyone to do is to use a password manager so they don't recycle their passwords.

Recycling passwords is how current threat actor groups like Gnostic Players, MABNA (currently on the FBI's most wanted list), and many others are able to access Office365 environments, Amazon S3 accounts, and private code repositories.

The approach is incredibly simple: they collect lists of credentials, target a company's admins, and engage in credential stuffing attacks until they gain access. The entire process can be automated and is extremely effective against organizations that do not have multifactor authentication enabled.

Lucky for us, threat actors are no different. In fact, chances are they will be younger, and possibly less experienced, which makes them more likely to recycle passwords across their accounts—especially their older accounts. Using the account and password information we discover within data breaches, we can actually begin tracking their accounts by looking for similar passwords.

NOTE Of course, there is always a possibility that a password will be *so generic* (e.g., “password123”) that trying to search for it would yield garbage results. More often than not, that will not be the case. If it is, just move on to another account.

Completing F3ttywap's Profile Matrix

Circa 2017, the Nulled.io hacking forum was hacked, and a complete dump of its MySQL database was circulated. It can be downloaded pretty much anywhere now, including sites like RaidForums.com.

The dumped database contained a full user list, which included fields like username, email address, hashed password, verification email address, and so on.

A search of the DB for the email address “kunt.x7@gmail.com” links us to member ID 525426. However, Kunt.x7 was not his primary email address; it was his backup address. We can see from the data that his primary email address is listed as “kunt.lsx@gmail.com.”

Boom! Now we have a new address to add to our matrix! Since we already covered going through the password reset in the previous chapter, let's move on to looking up passwords.

NOTE Important: The screens I am going to show you are from my Data Viper tool, which we will discuss later in this chapter. For now, I want to focus on the flow and techniques of researching passwords, then circle back to using a tool.

Running a simple email search across my Elasticsearch database (i.e., Data Viper) returns 21 matches for our new email address (Figure 18.1).

**Figure 18.1**

The first four matches give us two unique passwords and one password hash. The headings of the sections show the name of the data set where the information came from (e.g., the MPGH hack, Exploit.in, and Pemiblanco). Unfortunately, I had to block out the passwords. Instead, let's pretend that one of the passwords is "2119801m," which is actually very similar to the user's original password.

2119801m is a fairly unique password, which makes this a great example of one that we can search for further clues.

Searching for that password yielded 91 results! I am not going to post all 91 (because many are redundant), but let's look at the important ones and discuss why they are so relevant.

First, Figure 18.2 shows two emails (with different domains) associated with our password.



Figure 18.2

This could be a dead end. The username “m4l1h4ck3r” has not come up before but in this case is repeated quite a bit, so it is worth putting to the side.

Figure 18.3 is an account from the 000WebHost.com hack.



Figure 18.3

This is actually a massive find for a number of reasons. First (and most importantly), if you recall the verification questions from the previous chapter, we were able to piece together the following verification email string using a combination of Gmail and PayPal password reset clues:

98.*****op@gmail.com

Now compare that to the email associated with our password, and the similarities should become immediately apparent:

98.*****op@gmail.com

98.lukapop@gmail.com

Further, because the username on this account is m4l1h4ck3r, that also ties in any accounts with that username.



Figure 18.4

It seems our friend has a lot of accounts on 000WebHost.com. Figure 18.4 shows another account with the same password, and a different email and username.

This *could* be someone completely different. However, given that the word *Luka* appears in the verification email address, combined with the fact that the password is the same, there is a pretty good chance this is the same person.

Figure 18.5 gives us the final nail in the coffin for this person—again, another account opened on 000WebHost.com with the same password. This time we can actually see his full name, along with the kunt.lsx@gmail.com email address that we already knew about.

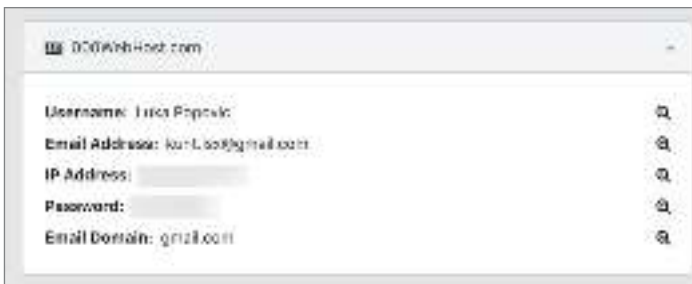


Figure 18.5

At this point it is just a matter of loose ends and gathering as much information as possible. We can use the name, IP, email addresses, and passwords to search for more accounts, which will begin to unravel all of his online movement.

An Important Wrong Turn

I mentioned this a few times already but want to say it again just in case: As far as I know, the *F3ttywap* used in this example has no affiliation whatsoever with *The Dark Overlord*.

Remember, experienced threat actors will reuse aliases belonging to other active threat actors. This was not an uncommon move for Cr00k (aka Ping).

This was an important lesson for me to learn—and one that I hope will resonate with you. Just because we have found the owner of *this set of F3ttywap accounts* does not mean he is the same F3ttywap associated with TDO. The important distinction is that none of these accounts could (in any way) be tied back to the F3ttywap on Siph0n's darkweb forum (where he and Arnie both advertised the sale of TDO's stolen medical data).

We will cover more examples of using recycled passwords to identify targets later in this chapter. First, let's take a step back and look at the process of how this data is collected.

Acquiring Your Data

Leaked data sets and password dumps are available everywhere. You can download them for paid “credits” on forums like [RaidForums.com](#), or you can even download *thousands* of data dumps of major sites like Adobe, Dropbox, LinkedIn, Matel, and Ashley Madison, from [www.databases.today](#). If you are looking to start your collection, this is where you will want to go to get caught up.

Assuming you get everything on [databases.today](#) and [RaidForums.com](#), that will probably get you to about the 50% mark in terms of what is actually circulating, privately.

If you need someplace to start, [RaidForums.com](#) is littered with people offering credential “collections” that include *billions* of username/password combinations. You can also harvest collections from sites like [pastebin](#). I personally stay away from collections and credential lists of any kind. The following note explains why.

NOTE The curation of the data in Data Viper took more than 18 months and, in my humble opinion, contains more quality data than I have witnessed in any other platform. I realize that is a bold claim, especially when the first statement you hear from every threat intelligence company is “we have more/better-quality data than anyone else.” The difference is, I'm not trying to sell memberships.

It's a hard statement to prove, sure, but I developed this tool as a means to achieve a very specific end, and I never had to sacrifice quality for quantity.

I am extremely proud of what I have built, and in the end, it was worth the time and effort because I was able to achieve my goal of identifying the members of TDO.

Data Quality and Collections 1–5

Sometime in early 2019, news of a massive collection of credentials caught the media's attention. Dubbed “Collection 1,” it included more than a billion

username/password combinations compiled from various data breaches and leaks.

In other words, if you already have the data from the actual leaks, then this collection will just be redundant data.

Troy Hunt, owner of Have I Been Pwned, wrote a lengthy blog post about the data discovered in Collection 1, as well as the data in Collections 2–5 (which he chose not to import into his database).

Troy and I were processing the data at the same time, and I did a fair amount of my own analysis for Collections 4 and 5 just to see if there was anything worth keeping.

Each data set contained roughly 10 to 20 different folders, each containing hundreds of randomly numbered files. My process for sanitizing the data was to first combine and de-duplicate the files within each folder. This would result in roughly 10–20 very large files of purely unique matches.

The next step was to combine and de-dupe each of those files until the result was only one master file for each collection.

Finally, I combined each of the two master collections files using a Linux de-dupe and merge command. The result was one very large file with a unique set of records from both collections.

Here are the statistics on the files and the outcomes of my analysis:

Collection 4:

- Original size—approximately 300 GB.
- Nearly 4 billion raw records (the largest of the Collections 1–5 releases).
- Each folder was de-duped and combined, resulting in a single master file of 60 GB and 1.8 billion entries.
- Comparing the new file with the original data showed that 55% of the overall release was duplicate.

Collection 5:

- Original size—approximately 220 GB
- 2.8 billion raw records
- After sorting and de-duping, the final file size was 45 GB and 1.2 billion unique entries.
- Comparing the results, 58% of the data in Collection 5 was duplicate.

Combining Collections 4 and 5

- Initial file size of both files totaled 105 GB with 3 billion raw entries.
- After the files were merged and de-duped, the final file size was 72 GB with just under 2.1 billion entries.
- Combining these data sets removed another 30% of duplicate entries.

Always Manually Verify the Data

At first glance, this sounds like a pretty decent outcome; 2.1 billion unique user-name/password combinations would be a terrific addition to any collection, right?

Unfortunately, this is where we have to take a moment to manually verify what we have. When we do this, we quickly come to the realization that the data is pure garbage.

The following is a few hundred entries randomly sampled from the final collection file. The columns contain email address and password:

"ddd-54000@hotmail.fr", "54000"	"ddd5403@mail.ru", "qwerty1123"
"ddd-54000@hotmail.fr", "cats654"	"ddd540422@yahoo.com.tw", "n888599"
"ddd-54000@hotmail.fr", "cats654"	"ddd5404@mail.ru", "lytrewq"
"ddd-54000@hotmail.fr", "Cats654540"	"ddd5404@mail.ru", "ddd5403123"
"ddd-54000@hotmail.fr", "ddd-54000"	"ddd5404@mail.ru", "qwerty"
"ddd-54000@hotmail.fr", "ddd54000"	"ddd5404@mail.ru", "qwerty"
"ddd-54000@hotmail.fr", "revoltec"	"ddd5404@mail.ru", "qwerty10"
"ddd54000@hotmail.fr", "revoltec"	"ddd5404@mail.ru", "qwerty1123"
"ddd-54000@mail.ru", "Cats65454000"	"ddd5405@bk.ru", "qwerty"
"ddd-54000@rambler.ru", "Cats654540"	"ddd5405@inbox.ru", "qwerty"
"ddd-54000@yandex.ru", "Cats65454000"	"ddd5405@list.ru", "qwerty"
"ddd-54001@mail.ru", "Cats65454000"	"ddd5405@mail.ru", "lytrewq"
"ddd-54002@mail.ru", "Cats65454000"	"ddd5405@mail.ru", "ddd5403123"
"ddd-54009@mail.ru", "Cats65454000"	"ddd5405@mail.ru", "qwerty"
"ddd5400@mail.ru", "qwerty"	"ddd5405@mail.ru", "qwerty"
"ddd5401@mail.ru", "lytrewq"	"ddd5405@mail.ru", "qwerty10"
"ddd5401@mail.ru", "ddd5403123"	"ddd5405@mail.ru", "qwerty1123"
"ddd5401@mail.ru", "qwerty"	"ddd5406812129@yahoo.com", "ddd54068"
"ddd5401@mail.ru", "Qwerty"	"ddd5406@bk.ru", "qwerty"
"ddd5401@mail.ru", "qwerty10"	"ddd5406@hotmail.com", "qwerty"
"ddd5401@mail.ru", "qwerty1123"	"ddd5406@inbox.ru", "qwerty"
"ddd5402@mail.ru", "lytrewq"	"ddd5406@list.ru", "qwerty"
"ddd5402@mail.ru", "ddd5403123"	"ddd5406@mail.ru", "qwerty"
"ddd5402@mail.ru", "qwerty"	"ddd5406@nm.ru", "qwerty"
"ddd5402@mail.ru", "qwerty"	"ddd5406@pochta.ru", "qwerty"
"ddd5402@mail.ru", "Qwerty"	"ddd5406@qip.ru", "qwerty"
"ddd5402@mail.ru", "qwerty10"	"ddd5406@rambler.ru", "qwerty"
"ddd5402@mail.ru", "qwerty1123"	"ddd5406@yandex.ru", "qwerty"
"ddd5403@mail.ru", "lytrewq"	"ddd5407906@163.com", "d5407906"
"ddd5403@mail.ru", "ddd5403123"	"ddd5407@bk.ru", "qwerty"
"ddd5403@mail.ru", "qwerty"	"ddd5407@inbox.ru", "qwerty"
"ddd5403@mail.ru", "qwerty"	"ddd5407@list.ru", "qwerty"
"ddd5403@mail.ru", "Qwerty"	"ddd5407@mail.ru", "qwerty"
"ddd5403@mail.ru", "qwerty10"	"ddd54088@mail.ru", "qwerty"
"ddd5403@mail.ru", "qwerty1123"	"ddd5408@bk.ru", "qwerty"
"ddd5403@mail.ru", "lytrewq"	"ddd5408@inbox.ru", "qwerty"
"ddd5403@mail.ru", "qwerty"	"ddd5408@list.ru", "qwerty"
"ddd5403@mail.ru", "Qwerty"	
"ddd5403@mail.ru", "qwerty10"	

"ddd5408@mail.ru", "qwerty"
 "ddd540900@gmail.com", "1472580369"
 "Ddd540900@gmail.com", "1472580369"
 "ddd5409@mail.ru", "qwerty"
 "ddd5409@sbcglobal.net", "96867340"
 "ddd5410@sian.com.cn", "19831014"
 "ddd@54154.ru", "drive330"
 "ddd54@163.com", "ninalove"
 "ddd5417@mail.ru", "qwerty"
 "ddd541983@mail.ru", "1645zzz"
 "ddd541984@mail.ru", "1645zzz"
 "ddd541985@bk.ru", "1645zzz"
 "ddd541985@inbox.ru", "1645zzz"
 "ddd541985@list.ru", "1645zzz"
 "ddd541985@mail.ru", "1645zzz"
 "ddd541985@rambler.ru", "1645zzz"
 "ddd541985@yandex.ru", "1645zzz"
 "ddd541986@mail.ru", "1645zzz"
 "ddd541987@mail.ru", "1645zzz"
 "ddd541@mail.ru", "45454546"
 "ddd541@mail.ru", "CFIEKZ1"
 "ddd541@mail.ru", "CFIEKZA"
 "ddd542931@hotmail.com", "mevemoza"
 "ddd542@mail.ru", "555555"
 "ddd542@mail.ru", "CFIEKZ1"
 "ddd542@mail.ru", "CFIEKZA"
 "ddd5431@bk.ru", "cfieksz"
 "ddd5431@list.ru", "cfieksz"
 "ddd5431@mail.ru", "cfieksz"
 "ddd54321d@bk.ru", "54321ddd"
 "ddd54321d@inbox.ru", "54321ddd"
 "ddd54321d@list.ru", "54321ddd"
 "ddd54321d@mail.ru", "54321ddd"
 "ddd54321d@rambler.ru", "54321ddd"
 "ddd54321d@yandex.ru", "54321ddd"
 "ddd5432@autorambler.ru", "cfieksz"
 "ddd5432@bk.ru", "cfieksz"
 "ddd5432@inbox.ru", "cfieksz"
 "ddd5432@lenta.ru", "cfieksz"
 "ddd5432@list.ru", "cfieksz"
 "ddd5432@mail.ru", "cfieksz"
 "ddd5432@myrambler.ru", "cfieksz"
 "ddd5432@qip.ru", "cfieksz"
 "ddd5432@r0.ru", "cfieksz"
 "ddd5432@rambler.ru", "cfieksz"
 "ddd5432@ro.ru", "cfieksz"
 "ddd5432srftged@ddd.com",
 "131531970"
 "ddd5432@yandex.ru", "cfieksz"
 "ddd54333@qip.ru", "555555"
 "ddd5433@bk.ru", "cfieksz"
 "ddd5433@list.ru", "cfieksz"
 "ddd5433@mail.ru", "cfieksz"
 "ddd5433@qip.ru", "555555"
 "ddd543@aol.com", "Thomas914"
 "ddd543@mail.ru", "555555"
 "ddd543@mail.ru", "cfieksz"
 "ddd543@mail.ru", "cfieksz@"
 "ddd543@mail.ru", "CFIEKZ1"
 "ddd543@mail.ru", "CFIEKZA"
 "ddd543@qip.ru", "555555"
 "ddd543@rambler.ru", "1q2w3e"
 "ddd543@rambler.ru", "555555"
 "ddd543s@mail.ru", "fdsa1210"
 "ddd543s@yahoo.com", "fdsa1210"
 "ddd543s@yahoo.com", "fdsa12101"
 "ddd543s@yandex.ru", "fdsa1210"
 "ddd543s@ya.ru", "fdsa1210"
 "ddd543@yahoo.com", "csh12345"
 "ddd543@yandex.ru", "555555"
 "ddd54434@mail.ru", "egor89227"
 "ddd54435@mail.ru", "egor89227"
 "ddd54436@mail.ru", "egor89227"
 "ddd5445d4kcjksd@
 qq.com", "82546004"
 "ddd5445@mail.ru", "fdffdfdf"
 "ddd54484848484dd@56.com", "198635"
 "ddd544@aol.com", "Thomas914"
 "ddd544@mail.ru", "555555"
 "ddd544@mail.ru", "CFIEKZ1"
 "ddd544@mail.ru", "CFIEKZA"
 "ddd54511884@yandex.ru", "qqqqqqqq"
 "ddd54542@yandex.ru", "54545454m"
 "ddd54544@mail.ru", "54545"
 "ddd545452@mail.ru", "6541236e"
 "ddd545453@mail.ru", "6541236e"
 "ddd545454@bk.ru", "6541236e"
 "ddd545454@inbox.ru", "6541236e"
 "ddd545454ksa@gmail.
 com", "6541236e"
 "ddd545454@list.ru", "6541236e"
 "ddd545454@list.ru", "ddd54545"
 "ddd545454@mail.ru", "6541236e"
 "ddd545454@mail.ru", "6545236e"
 "ddd545454@qip.ru", "6541236e"
 "ddd545454@rambler.ru", "6541236e"
 "ddd545454sla@web.de", "6541236e"
 "ddd545454sla@yahoo.co.uk",
 "6541236"
 "ddd545454@yahoo.com", "444film"

```

"ddd545454@yandex.ru", "6541236e"
"ddd545454zls1@yahoo.
com", "6541236e"
"ddd545455@mail.ru", "6541236e"
"ddd545456@mail.ru", "6541236e"
"ddd54545@mail.ru", "54545"
"ddd54545@mail.ru", "6541236e"
"ddd54546@mail.ru", "54545"
"ddd54546@mail.ru", "ddd54546@
mail.r"
"ddd@54546.ru", "drive330"
"ddd54547@mail.ru", "54545"
"ddd54547@yahoo.com", "sss54547"
"ddd54547@yahoo.com.tw", "sss54547"
"ddd545488@163.com", "1150718345"
"ddd5454@gmail.com", "255555"
"ddd5454@hotmail.com", "255555"
"ddd@5454.ru", "drive330"
"ddd5454@YAHOO.COM", "ddd789456"
"ddd54554545@juno.com",
"ddddgvlie6"
"ddd.54.55@hotmail.com",
"1234567894"
"ddd5456@mail.ru", "111qqqaaa"
"ddd5457@aol.com", "12696dd"
"ddd545@aol.com", "Thomas914"
"ddd545@mail.ru", "CFIEKZ1"
"ddd545@mail.ru", "CFIEKZA"
"ddd545y999@qip.ru", "3334444"
"ddd545y999@yandex.ru", "33344"
"ddd545y999@yandex.ru", "333444"
"ddd545y999@yandex.ru", "3334444"
"ddd545y999@yandex.ru", "33344441"
"ddd545y999@yandex.ru", "3334444Q"
"ddd545y999@yandex.ru", "545999"
"ddd545@yandex.ru", "1234567"
"ddd545@yandex.ru", "291979"
"ddd54608@mail.ru", "19a89a"
"ddd54615461", "54615461"
"ddd54615461@bigmir.
net", "54615461"
"ddd54615461@bk.ru", "54615461"
"ddd54615461@email.ua", "54615461"
"ddd54615461@freemail.
ru", "54615461"
"ddd54615461@gmail.com", "54615461"
"ddd54615461@gs.uz", "54615461"
"ddd54615461@hotmail.com",
"54615461"
"ddd54615461@hotmail.ru",
"54615461"
"ddd54615461@inbox.ru", "54615461"
"ddd54615461@i.ua", "54615461"
"ddd54615461@list.ru", "54615461"
"ddd54615461@mail.com", "54615461"
"ddd54615461@mail.ru", "54615461"
"ddd54615461@narod.ru", "54615461"
"ddd54615461@qip.ru", "54615461"
"ddd54615461@rambler.
ru", "54615461"
"ddd54615461@tut.by", "54615461"
"ddd54615461@ukr.net", "54615461"
"ddd54615461@yandex.by", "54615461"
"ddd54615461@yandex.
com", "54615461"
"ddd54615461@yandex.kz", "54615461"
"ddd54615461@yandex.ru", "54615461"
"ddd54615461@yandex.ua", "54615461"
"ddd54615461@yandex.uz", "54615461"
"ddd54615461@ya.ru", "54615461"
"ddd54615461@ya.ua", "54615461"
"ddd5461546@gmail.com", "54615461"
"ddd5468@yandex.ru", "23011985"
"ddd_5470ddd@yahoo.
co.jp", "19880428"
"ddd_5470ddd@yahoo.
co.jp", "87160044"
"ddd-5476@qq.com", "325221"
"ddd5478d@hotmail.com", "ddd8478d"
"ddd5478@mail.ru", "1234567890"
"ddd5479@yandex.ru", "1322455"
"ddd5479@yandex.ru", "k1322455"
"ddd547@bk.ru", "mama8962"
"ddd5551970@list.ru", "gjkrrjdjltw"
"ddd5551970@mail.ru", "5551970"
"ddd5551970@mail.ru", "ddd5551970"
"ddd5551970@mail.ru", "gjkrrjdjltw"
"ddd5551970@qip.ru", "gjkrrjdjltw"
"ddd5551970@rambler.ru",
"gjkrrjdjltw"
"ddd5551970@yahoo.com",
"gjkrrjdjltw"
"ddd5551970@yandex.ru",
"gjkrrjdjltw"
"ddd5551971@mail.ru", "5551970"
"ddd5551971@mail.ru", "ddd5551970"
"ddd5551971@mail.ru", "ddd5551971"
"ddd5551979@mail.ru", "5551970"
"ddd5551979@mail.ru", "ddd5551970"

```

"ddd5551979@mail.ru", "ddd5551979"
 "ddd5552008@yandex.ru",
 "t30u87l89a"
 "ddd.5552010@yandex.ru", "zyrjdf"
 "ddd.5552010@yandex.ru", "Zyrjdf"
 "ddd.5552019@yandex.ru", "zyrjdf"
 "ddd555222554@mail.ru",
 "321123321d"
 "ddd555222555@mail.ru",
 "321123321d"
 "ddd555222556@mail.ru",
 "321123321d"
 "ddd555222@mail.ru", "28111988"
 "ddd5552@hotmail.com", "947600"
 "ddd.5552@mail.ru", "11aa11"
 "ddd55533443@mail.ru",
 "ddd555444333"
 "ddd55533444@mail.ru",
 "ddd555444333"
 "ddd55533445@mail.ru",
 "ddd555444333"
 "ddd55533@aol.com", "8cwxkq43"
 "DDD55533@AOL.COM", "cairo"
 "DDD55533@AOL.COM", "dav2DAVID"
 "ddd55533@aol.com", "david"
 "ddd55533@aol.com", "DAVID"
 "ddd.5553@mail.ru", "11aa11"
 "ddd.5553@mail.ru", "159753waTsOn"
 "ddd5553@mail.ru", "ddd777"
 "ddd5553@tianya.cn", "331769638"
 "ddd55544@mail.ru", "inafngjn"
 "ddd55545@hotmail.com", "035611295"
 "ddd55545@mail.ru", "inafngjn"
 "ddd55546@mail.ru", "inafngjn"
 "ddd55547@hotmail.com", "0876585486"
 "ddd.5554@bk.ru", "11aa11"
 "ddd.5554@gmail.com", "11aa11"
 "ddd-55.54@gmx.de", "qazxcv"
 "ddd.5554@inbox.ru", "11aa11"
 "ddd.5554@list.ru", "11aa11"
 "ddd.5554@mail.ru", "11aa11"
 "ddd.5554@mail.ru", "159753waTsOn"
 "ddd5554@mail.ru", "89222555920"
 "ddd-55.54@mail.ru", "ddd-55.54"
 "ddd-55.54@mail.ru", "ddd5555"
 "ddd5554@mail.ru", "ddd777"
 "ddd-55.54@mail.ru", "qazxcv"
 "ddd-55.54@mail.ua", "qazxcv"
 "ddd.5554@pochta.ru", "11aa11"
 "ddd.5554@qip.ru", "11aa11"
 "ddd.5554@rambler.ru", "11aa11"
 "ddd.5554@rambler.ru", "11aa111"
 "ddd.5554@rambler.ru", "11aa11123"
 "ddd.5554@rambler.ru", "11aa11a"
 "ddd.5554@rambler.ru", "11aa11qwe"
 "ddd.5554@rambler.ru", "Ddd.5554"
 "ddd5554@rambler.ru", "ddd777"
 "ddd-55.54@web.de", "qazxcv"
 "ddd.5554@yandex.ru", "11aa11"
 "ddd.5554@yandex.ru", "1212aa1212"
 "ddd55551@inbox.ru", "cfifcfif"
 "ddd55551@mail.ru", "121212"
 "ddd55551@mail.ru", "388211"
 "ddd55552@inbox.ru", "cfifcfif"
 "ddd55552@mail.ru", "5234ww"
 "ddd55553@inbox.ru", "cfifcfif"
 "ddd55554@bk.ru", "04111997az"
 "ddd55554@inbox.ru", "cfifcfif"
 "ddd55554@mail.ru", "cfifcfif"
 "ddd55554@rambler.ru", "27101993m"
 "ddd555550@mail.ru", "171000ddd"
 "ddd555551@mail.ru", "171000ddd"
 "ddd55555@2lco.com", "000000"
 "ddd555552@mail.ru", "171000ddd"
 "ddd555553@inbox.ru", "poiv4u9a"
 "ddd555554@inbox.ru", "poiv4u9a"
 "DDD555555555555@bk.ru", "44444444"
 "DDD555555555555@inbox.ru",
 "44444444"
 "DDD555555555555@list.ru",
 "44444444"
 "ddd5555555555@meta.ua", "225517"
 "ddd55555555@mail.ru", "mamam12"
 "ddd555555@bk.ru", "bekl2001"
 "ddd5555555@hotmail.com",
 "ryremalu"
 "ddd5555555@inbox.ru", "bekl2001"
 "ddd5555555@inbox.ru", "poiv4u9a"
 "ddd5555555@list.ru", "bekl2001"
 "ddd5555556@inbox.ru", "poiv4u9a"
 "ddd5555557@inbox.ru", "poiv4u9a"
 "ddd555555d@sina.com", "nijido"
 "ddd555555@gmail.com", "102030"
 "ddd555555@hotmail.com",
 "metiancai"
 "ddd.555-555@hotmail.co.th",
 "123456"
 "ddd555557@mail.ru", "06020602"
 "ddd555557@rambler.ru", "06020602"


```

"ddd55555@bk.ru", "04111997az"
"ddd55555ddd@mail.ru", "55555555"
"ddd55555ddd@mail.ru", "55555555"
"ddd55555ddd@yandex.ru", "55555555"
"ddd55555ddd@yandex.ru", "55555555"
"ddd55555d@hotmail.com", "larson07"
"ddd55555d@rambler.ru", "larson07"
"ddd55555@ibox.ru", "cfifcfif"
"ddd55555@inbox.ru", "cfifcfif"
"ddd55555@inbox.ru", "ddd55555"
"ddd55555@inbox.ru",
"ddd55555inboxrucfifcfif"
"ddd55555@inbox.ru", "fifcfifc"
"ddd55555@mail.ru", "55555"
"ddd55555@mail.ru", "cfifcfif"
"ddd55555@meta.ua", "225517"
"ddd55555@rambler.ru", "ddd77777"
"ddd55555@yahoo.co.jp", "ddd66666"
"ddd55555@yandex.
ru", "qwerty123456"
"ddd555566333@163.com", "211385"
"ddd55556@bk.ru", "04111997az"
"ddd55556@inbox.ru", "cfifcfif"
"ddd55556@mail.ru", "55555"
"ddd55556@mail.ru", "cfifcfif"
"ddd55557@inbox.ru", "cfifcfif"
"ddd55558@inbox.ru", "cfifcfif"
"ddd55559@inbox.ru", "cfifcfif"
"ddd-55.55@aliceads1.fr", "qazxcv"
"ddd-55.55@bk.ru", "qazxcv"
"DDD5555DDD@bk.ru", "EHtsvKWv"
"DDD5555DDD@inbox.ru", "EHtsvKWv"
"DDD5555DDD@list.ru", "EHtsvKWv"

```

The data is clearly manufactured. The emails and passwords are all virtually identical. It looks like this is more of a password guessing list than an actual combo of real data.

I can't imagine any of this ever being useful, which is exactly why I have never bothered to gather and process any collections data.

EXPERT TIP: TROY HUNT

One of the things that I see a lot of is credential stuffing attacks against Spotify. I regularly get people who say, "Hey, Have I Been Pwned, I just picked up a paste (from pastebin) and it looks like Spotify had a data breach." And then you go and look at the paste, and realize, "Wow, this is formatted just like every other 'data breach,' where it's actually got username and password in cleartext and then the type of membership, like family, premium, or what it is."

Every single time I'll go back to validate with some of the users, and it always comes back to bad reused passwords. I don't know if "garbage" is the word I'd use because they're legitimate credentials, but they're not due to "data breaches." They're just due to password reuse. Although, having said that, they are often passed off as data breaches with people saying, "Hey, we got this Spotify data breach." No, you don't.

On the theme of fabricated breaches, another one that I see a lot of people try to pass off is Twitter, where they'll say, "Hey, here's the 30 million accounts from Twitter." This always smells bad, and I actually reached out to the Twitter CISO once to confirm some of the data, and it always comes back as being part of another combo list of credential stuffing. It's just the same recycled passwords being passed around over and over.

Where to Find Quality Data

At this point, you might be asking yourself: OK, then where do I get good-quality data? Well, it's not a quick or easy process.

You can find some of your basic, public data sets on sites like `RaidForums.com` or `databases.today`. For anything more than that, it will require you to get involved in the underground data market.

I have provided a number of stories that included me creating fake personas, interacting with threat actors, and basically putting my entire life on hold while I devoted thousands of hours to embedding myself into this underground scene.

This isn't for everyone, but for those of you who get to do this as a full-time job, or for those who have the proper motivation (as I did with TDO), then know that this will not be an overnight process.

Data Viper

Data Viper (www.dataviper.io) is my tool. I developed it as a means to help identify members of The Dark Overlord. I did so because I could see a clear lack of necessary data throughout all of the threat intelligence tools that I tried—even the companies that refused to help me on this quest (and some were quite rude about it). I luckily had friends who were willing to search their data sets for me, and I quickly came to find out that the reason these companies were so protective of their data is that they didn't have any.

So I decided to collect my own, and thus Data Viper was born. The tool uses a mix of ReactJS and PHP for its GUI, and an Elasticsearch backend for the database storage. The concept for the tool is not overly complex, which is why I say anyone can build it. In fact, you don't even need to build your own front-end GUI; you can just use Kibana, which is already part of the Elasticsearch project. If you are interested, you can read more about Kibana here: <https://www.elastic.co/products/kibana>.

I have discussed Data Viper numerous times throughout this book, and earlier in this chapter I provided an example of how the tool could be used to uncover the identity of a threat actor by searching for common passwords located within data breaches. In reality, any tool could be used to do this. In fact, you technically don't even need a "tool"; you just need access to the right data, and the ability to search that data (which is where Kibana comes in).

Later in this chapter I will offer a few different pre-packed solutions that can be used if you don't want to build your own. First, let's look at how the data was used to form a solid conclusion.

Forums: The Missing Link

Being able to search passwords is really just one part of Data Viper. When you are first starting an investigation, being able to search for passwords may not necessarily be that helpful because chances are you won't know which accounts or email addresses are associated with your threat actor, and you probably won't even have enough information on your target to know what you should be searching for.

I realized this when I first started building Data Viper, which is why I also took the time to build a network of complex web scrapers designed to scrape every known hacker forum and paste site I could find. Now, instead of just being able to search for passwords, I also developed the ability to search for specific keywords across forum users, threads, and paste sites.

At the time of writing this chapter, I have over 11 billion records (independent of pastes or collections), and hundreds of millions of scraped forum posts.

Figure 18.6 shows Data Viper's home screen.

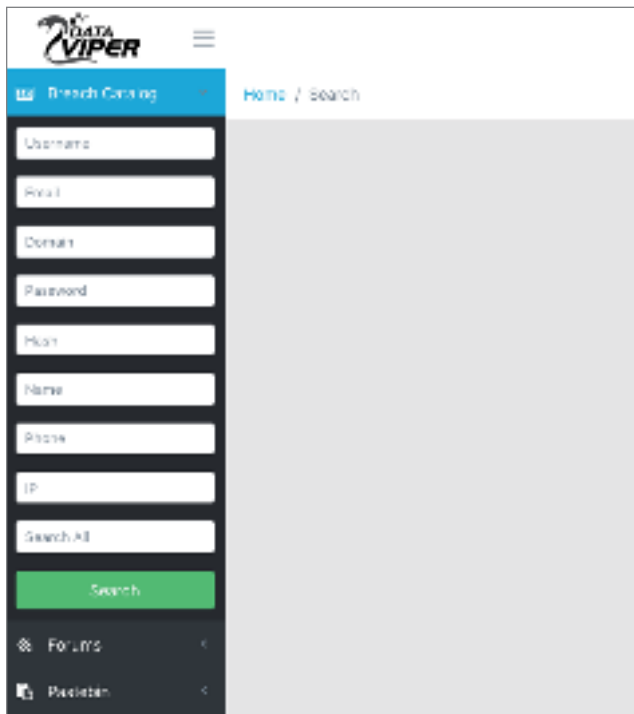


Figure 18.6

The UI is simple. I purchased a pre-built theme for the front end so I could just pop in components without having to code them. The fields shown in Figure 18.6 are the basic search elements that I typically look for and pivot with.

Identifying the Real “Cr00k”

The identification of cr00k was the first big break I had during the TDO investigation. Performing a simple Google search of cr00k will reveal a plethora of activity by this user on various clearweb hacker forums.

At first glance, tracking the whereabouts of this user *should* be easy, especially given the large trail he has left across the Internet. Every time I spoke with a researcher or intelligence company about this user, they all immediately and conclusively agreed that cr00k is “Zain M.,” a low-level carder from Saskatchewan, Canada.

Well, it turns out they are all right. Cr00k *is* a small-time carder from Canada, and all evidence points to Zain being cr00k.

Unfortunately, that Cr00k has absolutely no affiliation with TDO, and it took months to figure this out.

For the longest time, I was convinced—as was everyone else—that the Cr00k who openly talks about carding on the clearweb must be the same Cr00k that is affiliated with TDO.

There was just one nagging problem that I could not resolve. Morpheus says it best in the first *Matrix* movie: “It’s like a splinter in your mind, driving you mad.”

Figure 18.7 is a screenshot of one of the original messages selling posted by cr00k on Exploit, a Russian hacking forum.



Figure 18.7

The important thing to notice in this screenshot is the contact address, cr00k@swissjabber.ch.

This was the crux of the problem. Every password, email address, and online resource linked back to our friend Zain as the man behind Cr00k. Yet nothing, not one single shred of information, linked Zain to the XMPP address cr00k@swissjabber.ch.

Enter Data Viper.

Tracking Cr00k's Forum Movements

Because I took the time to access and scrape as many darkweb and clearweb forums as possible, I could essentially track cr00k's movements. Most of the clearweb matches were no good in this case since they ultimately led to our friend Zain. However, we could easily match the user Cr00k from Exploit to the same Cr00k that liked to hang around KickAss and 0Day (two other darkweb hacking forums) because of his contact address (cr00k@swissjabber.ch). Figure 18.8 is a screenshot of a thread on KickAss by Cr00k.



Figure 18.8

Since the Jabber IDs are the same, we can safely conclude that we are dealing with the same person. Now, using Data Viper, we can run a list of all his posts across KickAss and other forums (sample of posts shown in Figure 18.9).

Timeline Analysis

Being able to view all of Cr00k's posts across relevant forums in a data-ordered fashion like this allowed me to see the following chain of events (to make things easier for you, I will show the events in screenshots rather than small cutouts from Data Viper).

Figure 18.10 shows that on March 29, 2016, user Cr00k posted a sales thread on the KickAss forum where he was selling a number of hacked databases (and asked to be contacted at cr00k@swissjabber.ch).

author	forum	message
cr00k	KickAss	DO which I was selling Mother [REDACTED] went to the press...
cr00k	KickAss	If they are high quality then they can be used for online PoS systems like Strife if you disable CVV number authorization/checking. Good luck with sales!
cr00k	KickAss	Prevent leaks by: - background checks, as you see the idiot who used the same PGP keys. This incident could of been easily prevented if you think about it right now. - allow people only if they either a) offer a service and will be active on the board (e.g. discussion)
cr00k	KickAss	1. EC with loads of traffic 2. Phishing 3. E-mail spam with Word/PDF exploits.
cr00k	KickAss	DK is dead, as in not coming back.
cr00k	KickAss	Why would you ever see a source code from an OS from 8 years ago? Anti-virus companies are maybe updating their methods, techniques and datasets <i>see</i> everyday. If you want to effectively test your malware/exploits <i>ago</i> fact OS's you should set-up a VM with the OS in particular (Kaspersky) on hardest mode.
cr00k	KickAss	Bitcoin mining is dead, better do ransomware with low price to encrypt on. 50 USD.

Figure 18.9

List of DB's for sale:



Feb 2016, 12 users

Carding/Melissa Feb 2016 full DB includes everything, 177k users vB history & private messages.

AmyForceOnline.com game network Feb 2015, 2m users, HDG history.

TeamHostUsers USA porn network, some name/address/city, around 400k users & private posts.

forums.employee.com, vB history 434k users, dumped Feb 2016

DataHub.com forum, 118k users, dumped Jan 2016

BotOfLegends.com November 2014, 235k users

Jobber ID: cr00k@swisjobber.ch
No set prices, please offer me in Bitcoin.

Figure 18.10

One of the databases for sale included the usernames and passwords for TeamSkeet.com, a USA porn network.

Shortly after, user “NeoBoss” posted the same data for sale on Dream Market, a former darkweb marketplace (Figure 18.11).

The screenshot shows a Dream Market interface. At the top, it says "Dream Market" with a tagline "A hardily expanding online marketplace" and "Established 2013". Navigation links include "Shop", "Messages: 0", "Account: \$0.00", and "softnews". A shopping cart icon and "Logout" link are on the right. The main content area displays a listing titled "TeamSkeet.com Porn Network Database! Dumped March". The listing details include: Vendor: TheNeoBoss (0), Price: \$0.98 (\$400), Ships to: Worldwide, Worldwide, Ships from: Worldwide, and Escrow: Yes. Below this is a logo for "TEAM SKEET" with the URL "www.teamskeet.com". The "Product description" section states: "So recently I managed to breach TeamSkeet.com, the giant USA porn network. By purchasing this database, you will basically have free porn accounts for life, or you could sell them separately. There are several databases included with this purchase. This includes: Admin (login DIS, 227k users), password database, forum and so on. teamskeet.com/members/neoBoss/1 address is 0.0.0.0/100.0.0.0/100.0.0.0". On the left, there's a "Browse by category" section with links like "Digital Goods: 8804", "Data: 318", "Pirated: 88", "Digital Goods: 8804", "Drugs: 11791", "Drugs: Paraphernalia: 473", "Services: 616", "Other: 1747", and "Onion mirrors". On the right, there's a "Links" section with "Home", "Help", "Vendor application", and "Earn money", followed by a "B Exchange" table with various items and prices, and a "News" section with links like "Free money by finding bugs" and "Forum Released".

Figure 18.11

The post read, “So recently I managed to breach TeamSkeet.com, the giant USA porn network. By purchasing this database, you will basically have free porn accounts for life, or you could sell them separately.”

On March 31, 2016, an article posted on *Vice.com* described a hacker breaching a porn network and selling the data online (Figure 18.12).

Hacker Breaches Porn Network, Advertises User Data on Dark Web

The hacker, selling data under the handle **TheNeoBoss**, has gained control of the porn site **Team Skeet**, and claims to have data on over 230,000 customers.

Figure 18.12

This article included several direct quotes from NeoBoss where he specifically took credit for the hack and wanted to “publicly shame Team Skeet for their poor practices.” At this point, one could assume that Cr00k and NeoBoss are the same people.

Except on April 3, 2016, Cr00k posted an upset message on KickAss regarding a partner of his that “went to the press” with a database he was selling (Figure 18.13).

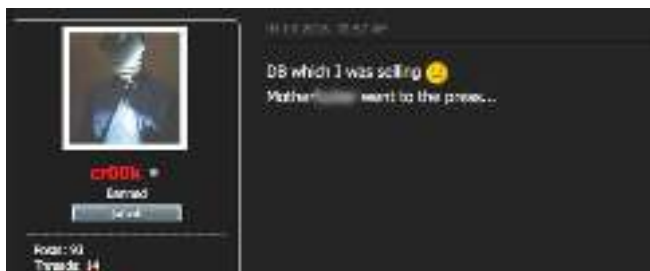


Figure 18.13

We already know from the previous post that he was selling TeamSkeet.com, so given the circumstances of events, it would not be wrong to assume that NeoBoss was his partner, and the “motherf***** that went to the press.”

It wasn’t until June 26, 2016, that user cr00k posted a “ripper report” for user w0rm on both the Exploit and Siph0n forums (Figure 18.14).

In the report, he lists several details about user w0rm (who happens to be a very well-known Russian scammer).



Figure 18.14

Again, notice the reference to “motherf*****” following the hash information. It would appear that it took cr00k several months to figure out that w0rm and NeoBoss were the same person.

NOTE Worth noting, the “ripper report” URL links to the same report on the Siph0n forum, posted under the name F3ttywap. That, combined with the several other identical TDO posts between Cr00k and F3ttywap on both forums are enough to conclude that these two threat actors are the same person.

Fast-forwarding to October 2016, an article on Motherboard (vice.com) states that user “Peace” (aka Peace of Mind) claims responsibility for hacking and dumping w0rm’s own hacker forum, w0rm.ws (Figure 18.15).



Figure 18.15

The Eureka Moment

All of these events seem fairly unrelated. Lucky for us, Peace was angry enough with w0rm to not only hack his database but publicly release it to the world.

As part of Data Viper, I have written scripts that will automatically strip apart database dumps like this—the user tables get pulled out and imported as user accounts, and the forum’s private messages, posts, and comments get pulled out and imported into the forums section.

In this case, though, simply importing this data would not have been enough. As a type of safety mechanism, I often manually search for usernames of interest whenever I receive a new piece of data. I have a practice of searching the entire data set *before* I strip apart and import the data—and the following clue will show you exactly why.

If I had not manually searched for “cr00k” in the w0rm dataset, I would have missed this crucial clue (which is why I’m sure so many others missed it as well).

Figure 18.16 shows the “announcement” table of the w0rm.ws database.

```
-- Dumping data for table 'announcement'
--
INSERT INTO 'announcement' ('announcementid', 'title', 'userid', 'startdate', 'enddate', 'pagetext', 'forumid',
'views', 'announcementoptions') VALUES
(1, 'Trusted Section', 'w0rm is under new management, as you can see there will be launched a section named
"Trusted" which only will be available to the most elite members and/or contributors.
Payment & info: ke7hb@w0rm.ws', -1, 365, 28)
```

Figure 18.16

The forum announcement (which would not have been picked up as a forum post or private message) states that “w0rm is under new management . . . payment and info: ke7hb@w0rm.ws.”

Looking further through the data, we can see that ke7hb was already a moderator on the w0rm forum.

Investigation into that username indicates that ke7hb was another one of w0rm’s usernames. So it would appear that he is trying to shift ownership of the forum to himself (just under a different alias).

However, the database changelog (Figure 18.17) tells us a very different story.

```
-- Dumping data for table 'userchangelog'
--
INSERT INTO 'userchangelog' ('changeid', 'userid', 'fieldname', 'newvalue',
'oldvalue', 'adminid', 'change_time') VALUES
(61, 387, 'username', 'ke7hb', 'cr00k', 188, 1463697364),
(62, 387, 'email', 'ke7hb@iranmail.com', 'cr00k@swissjabber.ch', 188, 1463697364)
```

Figure 18.17

Looking at the “userchangelog” table, we can see that the ke7hb account was renamed to cr00k, and the email address was changed from ke7hb@iranmail.com to cr00k@swissjabber.ch.

Boom!

Refer back to Figure 18.15: Peace claims ownership for w0rm hack. Peace *loves* attention from the media. He always has, and if he had not gone to the media claiming ownership for the hack, this would have been another dead end. Instead, now we can conclusively link cr00k with user Peace.

Vanity over OPSEC, Every Time

As I described earlier in this book when discussing Cyper, *Vanity always trumps OPSEC* (operational security). Never underestimate the willingness of a threat actor to be careless in the name of fame or notoriety.

To top this all off, Figure 18.18 shows a post by Cr00k on KickAss discussing w0rm's new management.



Figure 18.18

He was bragging, plain and simple.

Why This Connection Is Significant

Linking Cr00k to Peace is significant because user Peace can also be linked to “Ping,” the former owner and admin of Hell Forum. There is quite a bit of published information about the identity of Ping (most of which is incorrect) and Hell Forum. The important point is that I have now associated a member of TDO with Ping, the owner of the now-defunct Hell Forum—which means there is a good chance that he was still rolling with some of his old homies.

This shifted the *entire* focus of my investigation from researching current TDO-related events to digging up as much information as I could about Hell Forum and its members.

As it turns out, I was right. Ping and his buddy Revolt turned out to not only be thick as thieves online, but they also knew each other in real life (they are only 1 year apart in age, and live about 5 miles from each other).

NOTE Regarding my comment on the majority of published information on Ping being “incorrect.” In an absolutely brilliant move, several members of Hell Forum tried pinning the identity of Ping onto someone else by creating fake internal drama within the group and ultimately “doxing” the wrong person (Dimitri Barbu). This set off a chain reaction that led to the arrest of this person and several threat intel reports being released, which *confirmed* the identity of Ping to be this fake person.

To his credit, there was one article on Motherboard by Joseph Cox where he was able to uncover the truth behind the story. Once arrested, Dimitri gave up the real identity of Ping. Unfortunately, he was only 15 at the time, so those records were sealed and his name was never released.

These kids put on a dog-and-pony show that was, frankly, an amazing display of how powerful a single blog post can be in manipulating the perception of investigators and the news media.

To this day, TDO’s tactics have not changed. They regularly speak with reporters. In fact, a large majority of their stories come directly from author “Dissent,” who runs a blog at www.databreaches.net. In a direct conversation with her, Dissent told me that she has logged over 1,000 hours chatting with TDO directly. She regularly relays the information they provide her through her stories.

The point I am trying to make is that you should not always believe what you read, especially if the source of the information is coming directly from the targets you are investigating!

Starting Small: Data Viper 1.0

When I first started this quest, I was accessing all sorts of information on various forums and saving everything manually (literally using “Save As” to save each important page I came across). The result was a mess of HTML files that I needed to be able to reference. That, combined with all of the data sets I was collecting, turned into a nightmare for manual “grep” searching.

I started investigating different solutions, and before building a full Elastic-search database, I stumbled across DEVONthink, an amazing search tool for Mac. DEVON is available at www.devontechnologies.com.

DEVON is a data collection and indexing tool. Drag data into it, organize it by folders, and you have a pretty sweet search tool.

My DEVON database still contains all of the old data that I was able to scrape together. Figure 18.19 shows a search being run for “cr00k” just within the KA (KickAss) folder.



Figure 18.19

The license cost for DEVON is not expensive compared to the cost of building your own application. If you need a way to easily search through your own records, this is a pretty good tool. Unfortunately, this is only for Mac. If you are a Windows or Linux user, there are plenty of options available; dtSearch is a great tool used by many people (www.dtsearch.com). Another good option is docfetcher (docfetcher.sourceforge.net). You will need to explore which solutions are right for you.

Summary

This chapter discussed how hacked databases and password lists can be used to find new information on your targets. A real example of user F3ttywap was used to help finish the threat actor matrix that we have been building from the previous chapters. Since password reuse is so common among all people (even threat actors), you should always try to create new associations by looking for other accounts using similar passwords.

This chapter also took a deeper dive into Data Viper (www.dataviper.io) and how information from clearweb and darkweb forums can be used to trace the movements of threat actors and in the right context can ultimately lead to their identification. Since Data Viper is a homegrown application, this chapter also discussed ways in which you can build your own using freely available tools and data sets.

Interacting with Threat Actors

This final chapter details why it usually pays off to roll up your sleeves and jump in the muck with the threat actors you are investigating. You will be incredibly surprised (as I was) at how much information they will unknowingly give up.

Drawing Them Out of the Shadows

I had just returned home from the final DerbyCon, where I had given a talk on “Hunting Cyber Criminals.” I was on my way to taking my daughter to soccer practice when I received an overly suspicious tweet.

I knew my talk at DerbyCon would catch the attention of certain people, which is why I specifically included The Dark Overlord in the talk description, and made sure to include their name in my promotional tweets.

To take it one step further, I intentionally asked Adrian Crenshaw (and DerbyCon) to not publish the video of the talk for this exact reason. I had a suspicion that people close to the groups would catch wind of my talk and want to hear what was said.

The plan worked. A few days after my talk, I received the following tweet from a mysterious figure known as YoungBugsThug (Figure 19.1).



Figure 19.1

Sure, this could have been anyone. But then I received the following private message on Twitter from the same user (Figure 19.2).



Figure 19.2

Why would someone be asking me about Peace (Peace of Mind)? Peace hasn't been active since 2017. That *really* got my attention. So I started investigating YoungBugsThug and why he might be asking. The results turned out to be pretty monumental . . .

But before we go any further, I would like to take a step back and add some context behind why this is relevant.

Who Is WhitePacket?

A lot of the examples in this book focus on WhitePacket and Christopher Meunier. Throughout this book, I mention that he is a fellow security researcher and owner of his own security firm, White Packet Security, but I never actually explain why I chose to talk about him.

If you have read the official TDO report, then you already understand why I believe Meunier is nsa@rows.io, and the main persona behind The Dark Overlord.

The following is a personal account of the events leading to my conclusions, which will also set the stage for the significance of the identity of YoungBugsThug.

The Bev Robb Connection

Earlier in the acknowledgments, I pledged my eternal gratitude to Bev Robb. After I discovered that Cr00k was Ping, the original admin of Hell, I went on a quest to find some of his old associates. I reached out to reporters that wrote about Hell forum, and she was one of them.

Bev and I spent a lot of time talking, and she introduced me to someone that she knew from the forums. His name was Christopher Meunier, and he also went by the name WhitePacket.

Bev introduced Chris as a very talented white hat hacker, who was once a black hat that left the scene under difficult circumstances. Chris and Bev had lengthy conversations about Hell and why he chose to hang up his black hat, and she thankfully saved them all!

Chris fed Bev very convincing stories about how Ping was trying to threaten his life for releasing source code to the ZIB trojan (a TOR-based botnet). It was all part of a very elaborate plot to try to pin the true identity of Ping onto someone else, and it worked.

Bev spoke very highly of Chris, so when I reached out to him I did so by offering a contract job as a web pen-tester. During our chat, Chris was *extremely* abrasive. Granted, I was a stranger, but when I typically offer someone work they tend to be a bit more open and thankful. He was not.

Here is a critical excerpt from our first conversation:

```
VT:                so how do you know Bev?
whitepacket:       sorry not interested in discussing that
whitepacket:       are you in the law enforcement industry?
VT:                there's no money in being in LE
whitepacket:       I'm sorry man but you sound like you're either LE or
                   a f***ing retard, probably the latter.
VT:                why am i a retard?
whitepacket:       you and your friends NightCat/jasonvoorhees/hafez asad
                   and other d**kheads can go climb a wall of d**ks
VT:                NightCat?
VT:                jasonvoorhees?
whitepacket:       s u c k a d * * k
```

Little did he know that with one single word, he cracked open the entire investigation for me: NightCat.

Stradinatras

What Chris didn't realize is that NightCat was one of my aliases. In fact, I only used that alias one time, on Exploit.in, to communicate with a single person: Stradinatras.

Stradinatras also went as “Obfuscation” on the KickAss forum, where his logo was a picture of a white hat (I guess because he was masquerading as a white hat security professional). Obfuscation was very vocal on KickAss about not liking Arnie, the suspected head of TDO.

NOTE The names Stradinatras and Obfuscation were interchangeable—he would openly announce being both aliases, with the jabber address obbylord@jabber.de.

Interesting that he seems to use the word *lord* multiple times: ObbyLord, Dark Over Lord.

I reached out to Stradinatras on Exploit because I did not want him to know my alias on KickAss—so I made up a new one: NightCat.

There was a specific post on KickAss where Obfuscation was openly bashing Arnie and NSA(@rows.io) (Figure 19.3), so I thought he might be a good person to speak to for new clues.



Figure 19.3

Little did I know that Obfuscation and NSA were the same person, and his posts were simply a way to deflect attention away from himself. In revealing the name Nightcat to me, Chris made a critical error: he revealed himself as Stradinatras and Obfuscation, and set the wheels in motion for the rest of the investigation.

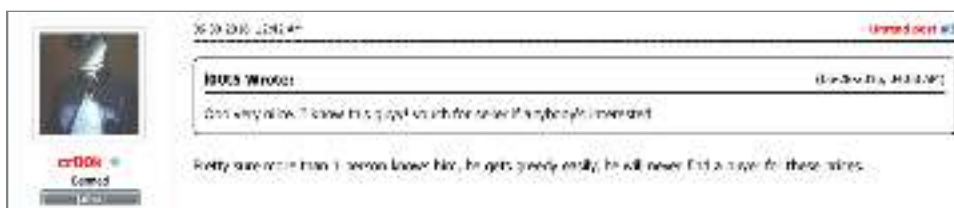


Figure 19.6



Figure 19.7

Remember earlier when he lumped in “NightCat” with JasonVoorhees? As it turns out, JasonVoorhees is *also* Arnie (a link that was made thanks to the marvels of stylometry analysis).

NOTE In a private conversation over Facebook Messenger, Wyatt reveals that the split between Arnie and NSA was a result of the Pippa Middleton hack.

After Wyatt’s arrest (and release), NSA became paranoid and assumed Arnie was now working with law enforcement. This would also explain why Obfuscation (aka NSA) called JasonVoorhees (aka Arnie) a dj*****.

The following response in this thread, written by l00t5 and directed at Obfuscation, is a gold mine (Figure 19.8).



Figure 19.8

Based on these conversations, it would seem that l00t5 (Arnie) and Obfuscation know each other well enough to call each other out. To me, it seemed obvious

l00t5 intentionally used the name Bill because he knew it meant something to Obfuscation.

Perhaps l00t5 knew it was his real name?

Who Is Bill?

Remember how I mentioned that Chris Meunier had a number of long talks with Bev Robb, and how fortunate I was that she saved them? The following screenshot is a piece of a seemingly random conversation where Bev asks Chris for screenshots (Figure 19.9).

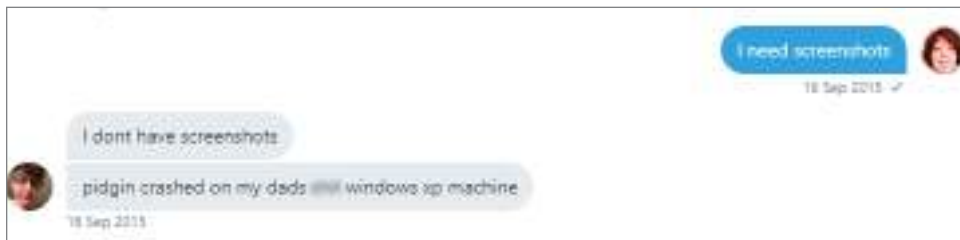


Figure 19.9

The important detail is that Chris uses his Dad's "windows XP machine" to run Pidgin. Next we have another screenshot from the following day where Chris pastes some code from the computer he is using (Figure 19.10).

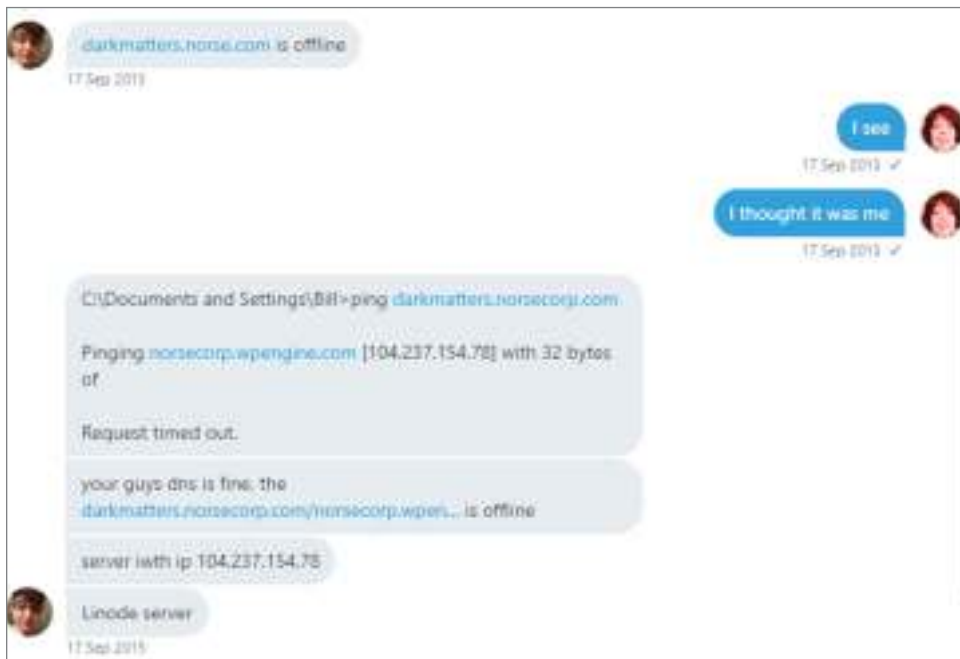


Figure 19.10

For those of you paying close attention, you will have noticed the username of Chris's computer is "Bill." Given the multiple times Chris mentions using his dad's computer, one could take a leap and guess that Chris's father's name is Bill.

So Who Exactly Is Bill?

William Meuneur is Christopher's father. What makes this even more interesting is that Bill is affiliated with a cybersecurity company in Canada. Chris' (former) LinkedIN page shows that he worked for a private cybersecurity firm, and the previous Figures show him openly using his father's machine to communicate with other people.

NOTE There is a substantial amount of other evidence linking these aliases, all of which can be read in the official TDO report. I just want to be clear in stating that I am not basing all of my theories on this one piece of information.

With that bit of background information, let's return to our story regarding the significance of YoungBugsThug (YBT).

YoungBugsThug

Now that we understand who Chris is and why we have been talking about him, let's get back to our mysterious YoungBugsThug. During our conversation, YBT asked me if I had ever used the handle Dr. X (Figure 19.11).

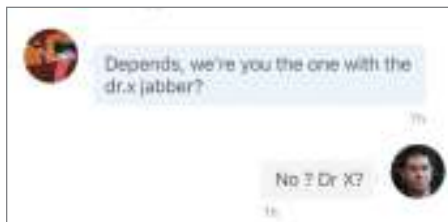


Figure 19.11

Honestly, no, I never have. This is not something I paid much attention to until it came up again the next day.

This time, he explained why he asked me. Unfortunately I can't get into precise details out of respect for the other person involved, but in a nutshell, YBT asked a particular reporter if I was Dr. X. I now had a third figure involved who I could approach to try and confirm identities.

I asked the reporter and mentioned the Dr. X scenario. I explained that I believed I knew who the person was, and asked how the reporter knew him.

The reporter did not say much, except that the person who mentioned Dr. X did so using his real name.

By this point I already knew I was talking to Chris. When I asked if this person's real name was Chris, the reporter was kind enough to confirm my suspicion.

How Did I Know It Was Chris?

There were a few things in our conversation that tipped me off, the first being that YBT was extremely over the top in trying to make me believe that he was Dennis K (aka Ping). He would say things like “how did you figure me out.”

Based on my conversations with Dennis, I don't believe he would ever say anything like that. The last thing he would do is admit that he knew anything about a person named Dennis, let alone that he *is* this person.

But the really big clue for me was the tone of the conversation. There was an actual point when we were texting over Signal where I got a chill because his tone gave me the creeps. Specifically, he would make a statement, then state my name right after. It had a very aggressive tone to it, like someone trying to show dominance. But he would also do this repeatedly, almost as if he was trying to set me up. Figure 19.12 is a perfect example.

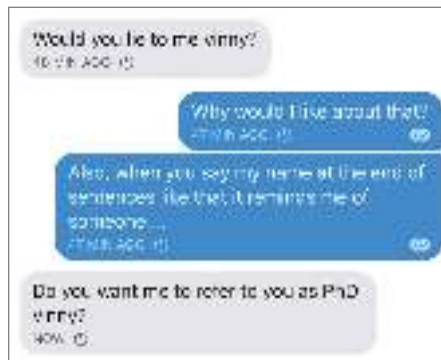


Figure 19.12

TDO was exactly the same way.

It is a really hard feeling to try to explain, but there were times when he and I would be having a normal conversation and he would throw in a sentence that would slap you back to reality and make you realize exactly who you were dealing with—someone that needs to feel superior to you in every way. It comes off him in waves and is unmistakable.

Here is an example from a conversation with TDO:

```
TDO: You white hats are idiots.
TDO: No one is dropping Xen vulns on you website, Vinny.
TDO: And after this, you'll need to let go of your self-admitted
      'obsession' of us. It's not healthy, Vinny. We're worried
      about you.
```

Plenty of threat actors know who I am and know they are speaking to me directly. No one has ever used my name. One could argue that this behavior was only directed at me and simply a coincidence.

The following is a private conversation between TDO and a redacted third party named James.

TDO: We understand your frustrations, James Perhaps the FBI raided you because you've divulged intelligence to us?

J: Only pogo and my wife know that the FBI asked about gift cards

TDO: You mean [Dissent], James She has a name, you know.

J: Oh.. yeah I tweeted it...

J: the gift cards.. so.. no big deal.. my wife just reminded me

TDO: You've been a great help to us, Justin. We owe you some internet money.

J: Sad, you owe the doctors and patients actually man

TDO: Tsk Tsk, James. You've been communicating with too many people outside of your region.

J: you are the weirdest person on earth

TDO: You'd have a lot of fun with a change of the hat, mate.

J: No I wouldn't

TDO: James, you've become quite a bore. What happened to that fun loving character we knew?

It would appear the language is fairly consistent.

There was also one more clue I came across. Let's call it the icing on the cake.

A Connection to Mirai Botnet?

Sometime in January 2019, a UK court sentenced Daniel Kaye, aka BestBuy aka Poporet, and owner/admin of TheRealDeal darkweb marketplace, to two years, eight months in prison for his connection and use of the Mirai IoT Botnet. According to several articles, Kaye is one of many people to have downloaded the source code of the Mirai malware when it was first published online in October 2016.

Fast-forward to March 2019, when Catalin Cimpanu, a cybercrime reporter for ZDNet, posted the following tweet regarding Kaye and the Mirai botnet. His post indicated that police seized Skype logs detailing conversations with two other Mirai co-conspirators, one named Chris (Figure 19.13).

How interesting that our friend YoungBugsThug went so far as to ask Catalin if Chris was from Canada.



Figure 19.13

I have not seen the recovered information, so I do not have any conclusive evidence that Chris (NSA/TDO) is the same person discussed in the Skype logs. I am basing this connection on the fact that that our Chris was also an admin of TheRealDeal marketplace (under the alias Peace of Mind), which was owned by Daniel Kaye. The two have a long and very traceable history working together going back to Hell forum.

The following is a piece of a conversation between Dennis K (whereami) and Chris discussing their involvement with BestBuy.

```
whereami: Bestbuy is getting ducked
whereami: Like back in UK and faces so many charges
whereami: I swear his is TRD or they arrested TRD to seize all my
           money
whereami: Eitherway all 3 of us f***ked
whereami: Let's say trd is not bb that means he was arrested over
           a year ago
whereami: That case is sealed then
```

YBT and I eventually moved our conversations to Signal, where he offered me several high-profile databases in exchange for confidential intelligence—specifically sealed indictment records—on Daniel Kaye and information pertaining to whether he would be extradited to the United States (Figure 19.14).

Finally, and my personal favorite, he flat-out asks me if I think he will be indicted (Figure 19.15).



Figure 19.14



Figure 19.15

As stated previously, this section is somewhat speculation given the existing relationship between BestBuy (Daniel) and Peace of Mind (Chris). There is more than enough evidence to show that the two worked together on other hacks and projects—TheRealDeal marketplace being one of the largest, which closed with an impressive multimillion-dollar exit scam.

NOTE If anyone reading this happens to know anyone in Europol with access to Daniel Kaye’s chat logs, I hope this bit of information helps as to who “Chris” might be.

Why Was This Discovery So Earth-Shattering?

There was something that did not sit right with me. In Figure 19.16, YBT seemed to go on about why I believed Ping to be Peace of Mind.



Figure 19.16

I thought about this for a long time before realizing something really important: I never once questioned the link between Peace and Ping. I can't remember where exactly I read this, or why I made that association in the first place, but from what I remember most articles and reports I read had no problem associating Peace with Ping.

When I went back and looked, I could not find a single shred of evidence connecting the two. The reason this conversation was so significant is that up until this point, I believed unconditionally that Ping and Peace were the same person.

As it turns out, I was wrong.

This was a huge slap in the face. Why had I gone on for so long assuming these two people were connected? I even reached out to some friends at other threat research firms and asked if they knew why these two aliases were connected—no one could find a trace of information in their records.

Question Everything!

When I went back and looked at all the evidence, I actually had a much easier time associating Peace of Mind with NSA, and in doing so actually closed a few

other loose ends about other aliases. Had YBT not contacted me out of fear of what I said in my talk, I would have gone on believing that Peace of Mind was actually Dennis K (Ping) and not Chris.

Since these two individuals have so many shared aliases between them, it may not actually matter in the eyes of the law—only that they are intricately interconnected. But from a research perspective, this experience taught me a very valuable lesson: *question everything*.

Never rely on assumptions, no matter how concrete the information or how many other people (or security researchers) are telling you the same thing. If you can't connect the pieces yourself, then you can't assume it to be true.

Establishing a Flow of Information

For the most part, I communicate with threat actors using various aliases. However, certain threat actors will still talk to you even if they know who you really are. There can be a number of reasons for this type of relationship to exist, but more often than not it is because you have something they want.

For the past six months, I have been speaking with actor Russian (aka Ping aka Cr00k, and one of the two people using the name NSFW) on an almost daily basis. He knew who I was from the beginning because I told him when I first approached him, citing the Krebs article about me as an excuse to talk to him.

Over time we established a strong rapport. I have no doubt that he has been speaking to me in order to keep tabs on what I know, and also to feed me disinformation when the opportunity arises. Like any good threat actor (and he is definitely in the top tier), he knows that in order for his disinformation to be effective, it needs to be sprinkled in with a substantial amount of real information.

Think of it like this: If he told me nothing but lies from the moment we began speaking, I would never trust anything he says. On the other hand, if he feeds me minor (but legitimate) information for months, he can then throw in lies that are relevant to him that I should naturally trust.

Once a relationship like this is developed, there is strong flow of information that can be an incredibly successful way to collect information on threat actor groups outside his circle. The less he cares about the people (or the further away from him they operate), the more likely he will be to completely open up and tell you what he knows.

As with anyone, if you can get them charged about a topic, they will reveal a lot of important information in the heat of the conversation. And given the frail interpersonal dynamic of threat actor relationships, it's often easy to find them in situations where they are feuding, or even to help spark the feud itself.

Leveraging Hacker Drama

Another equally important tactic is to leverage the childish arguments that occur between groups. You have to be patient and wait for it, but it always happens. Actors will inevitably try to work together, and before you know it, someone will get upset and the entire group starts to retaliate against each other.

This is an excellent opportunity to jump in and fan the flames. A number of threat actor groups hang out in similar Discord and IRC channels, so as long as you are monitoring and logging these channels you should be able to pick up the activity before they remove it.

The following is a recent example of internal drama that is occurring between GnosticPlayers and a few other people over Discord:

Sleepy	Btw Momondo, you know that I know who you are. I would keep your mouth shut because I don't really like keeping mine shut.
AmIEdgyEnough	@UnPirlaACaso who knows could still be him
UnPirlaACaso	nah:laughing
AmIEdgyEnough	Ye probably not
UnPirlaACaso	doubt the popo would care for him + he live in third world, he got no idea of what is FBI nor s**t like this ahahaha
AmIEdgyEnough	Looool
UnPirlaACaso	the FBI won't even be able to find him ahahahaha like wtf does this country relaly exist? and I would reach him just for this ahahaha
AmIEdgyEnough	Lmao yeah
liff	You just 69d that your partner lives in 3rd world country
UnPirlaACaso	ain't a secret he always said in public so doubt it would be a problem ahahaha
liff	ah ok then
Sleepy	Wouldn't be hard for anyone to find Momondo when pictures of his house next to McDonalds is indexed all over google
AmIEdgyEnough	Lmao, if you get him arrested then 100% there'll be like 3-4 other arrests lmao
Sleepy	Momondo already has been arrested He made a deal with USRS Once he came back I milked him for all the details.
AmIEdgyEnough	Ah yes
Sleepy	All I have to say about nclay is that he will forever have to worry about a heavy prison sentence. That's enough as it is.
AmIEdgyEnough	No he won't, autism has its perks

Sleepy	Are you autistic Photon?
AmIEdgyEnough	As well as the French judicial system
Sleepy	Does Poshmark know you're autistic?
AmIEdgyEnough	@Sleepy
	go ahead and let them know
	We'll see what happens

There is a lot of information that can be gathered from this conversation, but for my purposes, the most interesting piece was regarding Poshmark. Since I already know that Russian (aka NSFW) also used the alias Photon, and is responsible for hacking Poshmark (because he told me how he did it and shared a copy of the data with me), this conversation just connected him to a new alias: AmIEdgyEnough.

Was Any of That Real?

Another question you need to ask yourself, especially in situations like this, is whether what you are reading is real or staged. It would not take much effort for a group of these guys to get together and create a fake scenario; or a scenario that is more plausible is that the majority of the drama is real, but two friends have conspired to throw one piece of disinformation in the pot just to throw researchers off their TRAIL.

Since I know NSFW's history, I know that this is one area where he excels. As far back as I can trace his origins, he has been manipulating media and researchers with exactly this type of disinformation.

NOTE Worth noting, at this point in the timeline my ties with NSFW have been completely cut. After my Derbycon talk was announced ("Hunting Cyber Criminals") he began growing increasingly distant and we eventually stopped talking a few days before this all went down. Since he and I had been communicating using my real name, and because he knows I hang out in the same Discord channel, it would not be out of the realm of possibility that the information regarding NSFW and Poshmark was for my benefit.

There is no way to know for sure, so the only thing I could do is start to ask questions. When I did, I was told that AmIEdgyEnough approached someone with the following message, almost anticipating my reaction:

AmiEdgyEnough	BTW, if anybody asks where I'm going, say that my alias is Photon and that I'm going to enroll in a university so I won't be active. I'm sure when I go everyone will ask you first.
	Also, make sure to say I won't be active on raidforums.

```

My partner still will be active using the profile NSFW,
so continue to crack hashes if he asks, but don't tell
anyone photon is my alias.
If someone asks is Photon AmiEdgyEnough say no. If
someone says NSFW is AmiEdgyEnough lie and say yes,
just so people don't know that I'm photon but that
I'm someone else. I can pay you lmk.

```

The text contradicts itself—say I’m Photon, don’t say I’m Photon—but it is also clear that AmiEdgyEnough is trying to create a gap between NSFW and Photon.

Confusing as it is, when you break down this message with the understanding that Photon and NSFW are two different people, it actually makes perfect sense. He (AmiEdgyEnough) is asking this person to lie and say that he is not Photon, but if someone asks if he is NSFW, lie and say yes. The lie being that the current NSFW is his partner, not him. Deciphering this is enough to make anyone’s head explode!

Looking for Other Clues

Luckily, there were other clues that I could follow. AmiEdgyEnough and I had already been speaking, and he always seemed to go far out of his way to be an a*****.

After all this occurred, he apparently “caught wind” that I was asking about him and sent me a message telling me to stop asking around about him and to never contact him again.

Game on.

Sometimes, just talking to a person can give you the answers you are looking for, even if they aren’t really saying anything. During our last conversation, AmiEdgyEnough began to sound *extremely* familiar, but in a way I had not heard for some time:

```

XX:    Come on thought guy. What is it you think I did? Or maybe in your
       narcacistic world you think people are actually talking about
       you?
Ami:    "little narcacistic world" implies that I convey a sense of
       importance myself.
XX:    Yes, that is exactly what I am implying.
Ami:    However this is CONTRADICTIONARY to the fact that you've been asking
       about me MORE than my precieved narcissism.
Ami:    Well you can rest in peace knowing full well you are irrelevant
       in this, and nto who I am afraid, or consequently worried of.
       It is those you are
       speaking to that behold this information per se.
Ami:    Who unfortunately have made sure it is impossible or me to
       contact them.

```

XX: I need to go focus on my class now

Ami: And let's be honest, you're not going to be losing out my missing focus for this class of yours. Your f***ing degenerative brain cannot retain information for more than a day. So it's useless if you focus or do not focus. Because I do not speak like the fat oafs who are addicted to Discord for 24/7.

XX: You're right I'm just tired of talking to you.

Ami: That is why you feel tired, for not having the ability to decipher clear intricate messages.

Ami: Well if it wasn't n, then whoever your speaking to possesses no worthy information, unless they are in contact. So, on that note, you are free to use your underused brain.

Whoever this person is, it is clear that he is trying to make himself sound overly intelligent in the way he communicates.

That communication style reminds me of someone specific . . . and what a coincidence that it brings us right back to where we started.

Bringing It Back to TDO

Earlier in this book I mentioned there being two distinct TDO figures. TDO1 (aka Arnie) was the group lead in 2016, and TDO2 (aka NSA@rows.io) was the group lead from 2017 through 2019.

I have never had direct contact with TDO1 (since his tenure ended in 2016), but I have read a number of his chat transcripts and forum posts.

When TDO rebranded in 2017, they did so using the queen's English in an attempt to standardize all of their communications. Doing this allowed different people to act as representatives for the group.

In a private conversation with Wyatt, he mentions that TDO was really three core members—him, a “kid” who was good at hacking, and someone who was good with language and writing.

My first direct interaction with TDO was around October 2017. To this day, I can tell that the person I spoke to on this date was much more composed than the person I spoke to each time since. I always wondered who this mystery person was—and I knew it was not Arnie because he was in custody at the time in the UK.

The language similarity by itself would normally not be anywhere near enough to reach any sort of reasonable conclusion. However, since I already have more than enough evidence linking NSFW and Photon, to Cr00k (who was directly a member of TDO), it would not be a stretch by any means to think I was speaking with Cr00k on November 11, 2017.

Resolving One Final Question

There was one final nagging question that resolved itself when I began to consider that Cr00k was one of the people speaking under the alias of The Dark Overlord: Where is the money going?

Withdrawing Bitcoin

The following is an excerpt from a conversation with TDO regarding the withdrawal of their “earnings”:

```

1:55 AM      TDO      We earned almost 10M GBP last year.
1:55 AM      VT       thats impressive
1:55 AM      VT       how do you cash that out?
1:55 AM      TDO      We don't.
1:55 AM      VT       right
1:55 AM      TDO      That's the big secret.
1:55 AM      TDO      You want to see our addresses?
1:55 AM      VT       thats not much of a secret
1:55 AM      VT       i cant imagine many BTC exchanges offering to cash
                        out that much money

1:56 AM      TDO      I'll show you right now, since you're such an egit.
1:56 AM      VT       Sure. Show me.
1:56 AM      TDO      Of course not, we never cash out any of it.
1:57 AM      VT       SMH. you're not getting my point
1:57 AM      VT       If you cant cash it out, then the money is stuck
                        there

1:57 AM      TDO      It's not, that's what you're not understanding.
2:00 AM      TDO      Just a moment. Mind you, I'll sign these if need be.
2:01 AM      TDO      1EMWwmBJuvvES3eb51pNJnJEvD1RgQ7evA
2:01 AM      TDO      1NcgGFPT23KawMMp8vp1TKxVlqEitpPtdk
2:01 AM      TDO      Let' just start with these two.
2:03 AM      TDO      You're thinking we're unprofitable, and you're
                        wrong.

```

I always wondered exactly how TDO was able to cash out that much money. Another look into our pal Cr00k provides a very plausible scenario.

If I am correct, Cr00k, who I believe to be Dennis Karvouniaris (Instagram profile [dio_the_plug](#)), lives in Calgary, Canada. A simple Google search reveals that his father, is a prominent and award-winning chef.

Searching for the location of the business did not turn up anything useful. However, I learned another very valuable lesson: *Don't forget to search for the actual address in question, and not just the name!*

Once I did that, I saw something very interesting (Figure 19.17).



Figure 19.17

That's right. The location is equipped with its own Bitcoin ATM. What better way to anonymously withdraw your hard-earned extortion money?

I don't know if this is how the group was able to extract their extortion earnings, or if they used coin mixers, or some other method. I just find this to be a *interesting coincidence*.

Summary

This chapter highlighted some of my direct communications with different threat actors and explained why it can be necessary to directly engage them during an investigation. Sometimes it can be necessary to draw them out and have them come to you, and other times it may be necessary to start or fuel internal conflict between them in order to pair them against each other.

As I learned throughout this process, threat actors can unknowingly give away the greatest clues of their identity simply out of fear and self-preservation. Whatever your methods for communicating with them, be sure to take what they say with a grain of salt and question any information they present to you.

Cutting through the Disinformation of a 10-Million-Dollar Hack

There are days when you just wake up and something completely unexpected happens.

I thought I was finished with this book and then it just happened: GnosticPlayers, a well-known group that gained notoriety for a series of very high-profile hacks, decided to publicly post his/their involvement in the theft of 10 million dollars' worth of cryptocurrency.

Through an unprecedented admission of guilt, Nclay, the group's leader, decided to repent for his sins and publicly announce his involvement in the GateHub hack.

What came next was just as unexpected: the supposed tensions between the group members resulted in the public leaks of the group's private collection of hacked databases on RaidForums.com, subsequently referred to as "leaktober." Virtually every one of the group's high-value stolen databases, comprising billions of user accounts, passwords, and other private customer information, was published online and suddenly available as a free download.

Nclay continued to air his grievances over public forums, and with each new post came an additional piece of seemingly identifiable information. Not only was GnosticPlayers admitting his own guilt, but he had suddenly decided to partake in conversations where he would openly use his and his partners' real names and identities.

Within no time at all, both of the group's members had their full dox published, and Nclay, the group's leader, had dropped his aliases and was now referring to himself by his real name.

In the mix of all this confusion and excitement, one of my well-established contacts within the group's upper ranks revealed himself to me as a security researcher and investigator. This person was supposedly hired by one of the group's victims over a year ago to infiltrate and investigate GnosticPlayers in order to reveal Nclay's true identity. We were apparently on the same side. . . .

He quickly began feeding me information from his discoveries, which all seemed to confirm the information the group members were publishing about their own identities.

This was nothing short of spectacular! Attribution for all of these crimes could now be easily attributed to two boys living in France, all thanks to the alleged unstable mental condition of the group's leader.

Finally, on Monday, October 28, 2019, Nclay was apparently so remorseful for his crimes and the public betrayal of his friends that he decided to end his own life.

The story was now complete. The crimes had been accounted for, the data exposed, and the notorious hacker gone forever.

I call bullst.**

For more than a year, I have personally communicated with all members of GnosticPlayers, and I have grown to understand each of the members and their personalities, as well as their interactions with one another. I believe this entire scenario to be a carefully orchestrated show designed to send security researchers and law enforcement down a very well-lit path toward the wrong suspects.

In addition, *all* of Gnostic's databases have not been leaked. The leak, while extensive, was only partial, containing mostly older material. Newer databases like Quora, Epic Games (Fortnite), and many others did not make the cut, most likely because they are still extremely valuable. I also do not believe for one second that Nclay took his own life.

What I believe, and what I intend to show through the course of this chapter, is that these events have been carefully orchestrated by a person who is not only a silent member of GnosticPlayers but is also the same person who helped orchestrate the dox of Ping (and framing of Dimitri Barbu) discussed in previous chapters of this book.

GnosticPlayers

Circa February 2019, the name GnosticPlayers was born by way of several high-profile databases that went on sale on Dream Market (a darkweb marketplace). Some of the hacked databases included MyFitnessPal, MyHeritage, EyeEm, 8fit, and WhitePages.

Almost 1 billion stolen data records had now gone on sale, and this was only the group's first fire sale, dubbed "round 1."

While this was my first introduction to the group GnosticPlayers, I had already been speaking with the group's members for several months. Over the course of the next six months, there would be five more flash sales of freshly stolen databases.

The group was composed of two core members: Nclay and DDB (and a third, NSFW, which would be revealed later). Nclay was the hacker and DDB was the seller.

The flash sales were the result of internal tensions between the group members. Based on my conversations with the two members, Nclay decided to betray DDB and put all of his high-profile databases for sale.

The following is a private message sent to me by Nclay:

```
outofreach:   DDB hacked everything, took credit & made himself
               a name while he ain't s**t
outofreach:   because he took credit for hacks he didn't do and
               laundered the money for so long
outofreach:   he now feels he is good
```

Shortly after the first round of databases went up for sale on Dream Market, several of the databases were shared with Troy Hunt's HaveIBeenPwned website—according to information posted on the site, "by a source who requested it to be attributed to Kuroi'sh or Gabriel Kimiaie-Asadi Bildstein."

At the same time, Nclay's dox (personal information) was being leaked to a small group of people (myself included), naming him as Kuroi'sh, or Gabriel Bildstein.

NOTE Gabriel Bildstein aka Kuroi'sh is a known hacker previously arrested by French authorities in connection with the Vevo hack that defaced "Despacito" and several other YouTube music videos.

Around April 2019, the data from Bukalapak, an Indian e-commerce website, was also leaked to Troy Hunt. This time, however, the HIBP website stated, "data was provided to HIBP by a source who requested it to be attributed to 'Maxime Thalet'."

Months later, Maxime Thalet-Fischer would be named as the identity of Nclay's partner, DDB.

The conflict escalated for many months, ultimately leading to a public post by "Gnosticplayers" on the XRPchat.com forum, where he (allegedly Nclay) admits to hacking GateHub and being part of the team responsible for stealing 10 million dollars in cryptocurrencies (Figure 20.1).

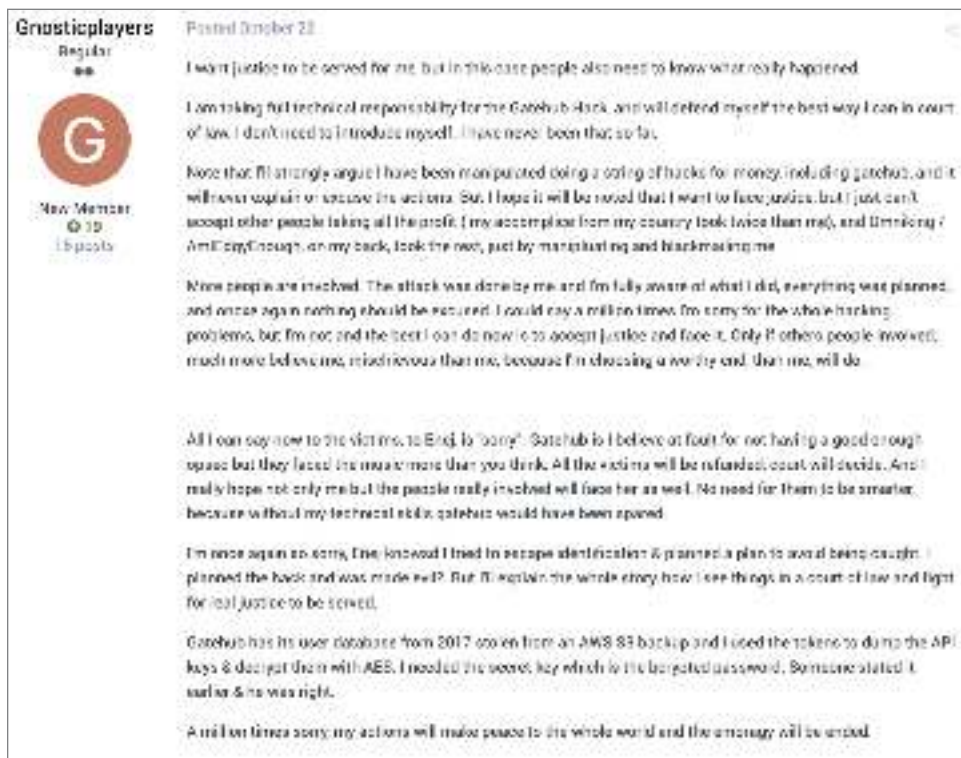


Figure 20.1

At the same time, the same person began leaking several of GnosticPlayers' hacked databases—offering them completely free on popular database dumping site RaidForums.com. Nclay used Raid Forums to air his grievances with other hackers and to also post the alleged name of his partner DDB as Maxime Thalet-Fischer.

Finally, after four days of chatter and confusion, the word began to spread that Nclay felt so bad for posting these messages and leaking these databases that he took his own life.

Tragic, yes. So before we analyze this information, here is some additional background information on the group and their known hacks.

Don't worry! Nclay didn't actually die. He is alive and well, and even makes a miraculous comeback later in this chapter to offer up his own commentary on how he was able to hack into all of these sites.

Sites Hacked by GnosticPlayers

The following is a list of sites hacked by GnosticPlayers (Nclay/DDB). This is not an exhaustive list; it is only a list of the sites that I have been made aware of.

- | | | |
|-------------------------|-----------------|-----------------------|
| ■ 500px | ■ Evite | ■ PromoFarma |
| ■ 8fit | ■ EyeEm | ■ Quora |
| ■ 8Tracks | ■ Fotolog | ■ Roadtrippers |
| ■ Animoto | ■ GameSalad | ■ Roll20 |
| ■ Armor Games | ■ GateHub | ■ ShareThis |
| ■ Artsy | ■ Ge.tt | ■ Storenvvy |
| ■ Avito | ■ GfyCat | ■ StoryBird |
| ■ BitMax | ■ HauteLook | ■ StreetEasy |
| ■ BookMate | ■ Houzz | ■ Stronghold Kingdoms |
| ■ Bukalapak | ■ iCracked | ■ Taringa |
| ■ Chegg | ■ Ixigo.com | ■ Wanelo |
| ■ ClassPass | ■ Jobandtalent | ■ WhitePages |
| ■ CoffeeMeetsBagel | ■ Legendas.tv | ■ Wirecard.br |
| ■ Coinmama | ■ LifeBear | ■ Xigo |
| ■ Coubic | ■ Mindjolt | ■ Yanolja |
| ■ Cryptaur | ■ Moda Operandi | ■ Yatra |
| ■ DataCamp | ■ MyFitnessPal | ■ YouNow |
| ■ Dubsmash | ■ MyHeritage | ■ YouthManual |
| ■ Edmodo | ■ Onebip | ■ Zomat |
| ■ Epic Games (Fortnite) | ■ PetFlow | ■ Zynga |
| ■ Estante Virtual | ■ Pizap | |

Gnostic's Hacking Techniques

The method for these hacks was revealed to me by NSFW (discussed in previous chapters) and confirmed by Nclay. The method was also confirmed by several of the hacked organizations.

The hack was simple but extremely effective: the group would target developers and would use recycled credentials to log in to their GitHub accounts. From there, they would search the developer's Git repositories for AWS keys or similar credentials that had been checked in. Once the hackers had the keys, they would log in to the company's systems and simply take what they wanted.

While this may seem like an obvious credential stuffing/account takeover attack, what is particularly interesting about Gnostic's method is *how* they were able to log into all of these GitHub accounts. The following is an excerpt of a

private conversation I had with Nclay via Twitter, where he kindly offered to explain his process so I could include it in this book.

```

VT:          how did you figure out how to log into those GitHub
              accounts without getting nailed with oauth ?
VT:          i didnt think you could just use user/pass to log into an
              api otherwise people would jsut brutefurce the f*** out
              of them
Nclay:       See the case of Capital one hacker
Nclay:       Github is getting sued for negligence
Nclay:       User/pass just kept working throughout the github api
VT:          thats amazing
Nclay:       And they tried to request a device confirmation after the
              gatehub hack
Nclay:       But the API was still allowing me to f**k
Nclay:       I didn't limit myself to github but it was fun. Once you
              know such flaw existed
Nclay:       See the zomato hack story
Nclay:       And see the 8tracks hack story
Nclay:       I hack gatehub mainly thanks to github
VT:          how were you brute forcing the github api? Just a python
              script?
Outofreach:  My own php script
Outofreach:  A small loop
Outofreach:  Could just be done with curl up user:pass
Outofreach:  And I was using my private dumps
Outofreach:  Canva which I hacked alone allowed me to f**k gatehub
Outofreach:  curl -u user:pass http://api.github.com/user
Outofreach:  -k -L
Outofreach:  you just needed to output it to a file after having
              parsed the Json ( can be done in one piping with jq)

```

It's simple, and it works.

NOTE Validating this is extremely simple, and as of December 2019, this still works on GitHub. To verify, simply run the following command from a command line:

```
curl -u username:password http://api.github.com/repos -k -L
```

If the credentials are valid, you will get a nice output of the user's private GitHub repositories. You can then use those credentials to check out any of the person's private repos.

What Nclay neglected to mention was how he was able to bypass GitHub's IP checks. For that, we can refer back to the following statement from NSFW in Chapter 2.

Once logged in I had to act quick to avoid Github's new ML algorithm to lock accounts out of using new IPs, so I immediately used ssh-keygen to add a new public SSH key to the user profile . . .

This is the part that was extremely creative. Once Gnostic was able to identify valid developer accounts by credential stuffing the HTTP-based API authentication, the group then added their own SSH keys to the developer's account using GitHub's command line tools (which were not subject to IP verification checks).

While this is not a "vulnerability," this oversight in GitHub's configuration is the reason why Gnostic Players has been so successful in hacking so many organizations, and the reason why Nclay feels he should be considered "one of the greatest hackers of all time."

Now that we understand how Gnostic was carrying out their attacks, let's get back to our story.

GnosticPlayers' Posts

The public admission of guilt by GnosticPlayers came as an obvious shock. Even more shocking is the fact that this person also decided to publicly leak so many of the group's private databases (making them public and freely available for anyone to download). This is where we will start our analysis.

Figure 20.2 is one of the initial posts made by GnosticPlayers1 on RaidForums.com, regarding the leak of the hacked Zynga database.

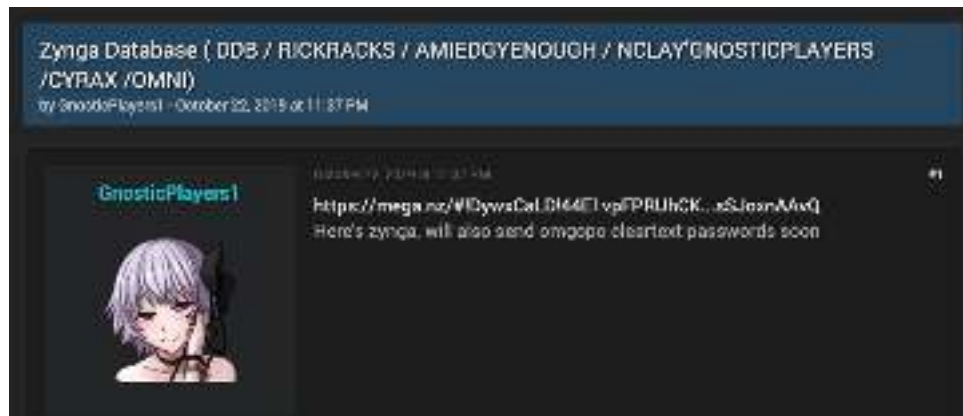


Figure 20.2

I believe this post was only designed to get people's attention for a few reasons. First, the Zynga link never actually worked; the mega.nz link was always suspended.

The person making these posts had no problem fixing the links for other leaked databases but would never repost Zynga. It stands to reason that if he was really interested in leaking all of the group's databases, he would have fixed

this one. I believe the reason he did not leak Zynga (and several others) is that they are still too valuable.

The following is a personal conversation between another researcher and DDB/Blinc:

```
blinc@jabber.ru:    it's gnosticplayer
blinc@jabber.ru:    but not selling it
xxx:                hook a brother up
blinc@jabber.ru:    i can't for real real i will not sell too forever
                    no one will have
blinc@jabber.ru:    i use it in fortnite
xxx:                what to hack other game accounts? and sell them?
blinc@jabber.ru:    yes
```

The post in Figure 20.2 is also interesting because the poster names RickRacks, AmiEdgyEnough, Cyrax, and Omni are all people who, as far as I know, have nothing to do with the Zynga hack.

Let's move on to Gnostic's other post (Figure 20.3), which contained the data for the BitMax crypto exchange and Over-Blog.com.

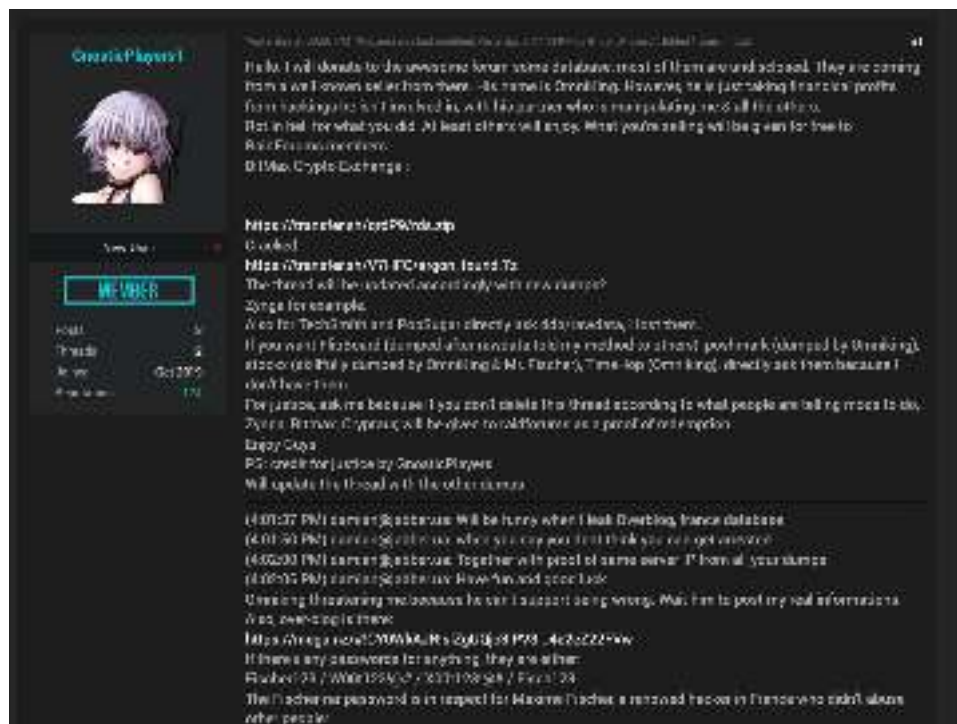


Figure 20.3

The part I found most interesting about this post was the sheer number of inaccuracies it contained. This is significant because the person posting this

message (allegedly Nclay) would have known that the information he is posting is incorrect. If he was going to go through the trouble of making a full confession, why intentionally include information that he knew to be incorrect?

Even more significant is *what* was incorrect. GnosticPlayers1 states that StockX, Poshmark, and Timehop were all hacked by Omniking, when in reality they were hacked by NSFW (this is irrefutable).

I also know with absolute certainty that Nclay knew this, as he and I have discussed it before. So the big question is why is this person protecting NSFW?

GnosticPlayers2 Emerges

On October 25, 2019, a new person emerges to RaidForums.com—called GnosticPlayers2—and posts the following message:

I'm writing this letter because I have to tell you the truth. There is currently a big misconception about me the real GnosticPlayers.

But to understand you must first find out the past. GnosticPlayers is not one person but two people behind the identity. Me that is writing this letter and GnosticPlayers1 who started this whole drama. To separate the two of us you can call me Gabriel and call GnosticPlayers1 for Nclay because it's our true identity.

I have known Nclay since 2015 and have always admired his skills. We have been very good close friends for many years. We met several times in real life because we come from the same country and we have nothing secret to each other. We knew everything about each other and we trusted each other.

Nclay was the mastermind behind all the hacks and executed them all on his own. But since this is all he knows he did not know what to do with the databases he hacked and how to brought money. It was my first idea was to sell it on the black market is when I first created the GnosticPlayers identity in the dream market.

My idea was to sell databases that nclay was hacking. He was very happy and loved this idea. He didn't once say no to that as long as he got the money he wanted. Also at this time we joined Rawdata / DDB his real name Maxime Thalet and helped us sell databases and gave us new ideas.

Everything was fine until Nclay broke through GateHub and stole \$ 10 million. Since we always shared money for everything before then Nclay was generous enough and gave me and Maxime 1/3 of each. We've all received about \$ 4 million each. But this is where the problem began.

Nclay became very irresponsible with his share of money and spent everything to buy 3 luxury cars for himself and gave a lot of money to his friends for free. But they were all fake friends who were just friends with Nclay because he was rich and had money and was exploiting him for money. After buying his cars and giving

the rest to his fake friend he had no money left. There he became greedy and began to regret his decision that he give Me and Maxim \$ 3 million each.

*He began asking us to give back his money or ask if he could have back it to a large extent. Of course since he was a friend of ours we gave him \$ 500,000 from both me and Maxime but he refused to accept the money because of his greed and said he was asking for half of it. He wanted us paid him back for \$ 3 million. We did not accept that and at that moment was angry and began to blackmail and threat for me and Maxim. If we don't do what he wants and return the money he gave us a pledge of friendship he will reveal our information to the public and f**k our lives.*

We did not listen to him and ignored him because we thought he was deceiving and would not do such a stupid thing. Since Nclay really has something wrong in his brain he really did. As noted he has published all the databases for free and voluntarily distributed my and Maximes information. He presents Me and Maxime as the main actors behind his actions while in fact the only major role in this case is himself only and none else.

He has published a lot of false and fake information that he was Gabriel but in real is actually me. He is posing as us using our information to get us in trouble as a revenge. He framed everyone except himself. He has also broadcast the names of many others who have played no part in the process or who have never been involved in anything simply to confuse people and spread false information. His words should not be reliable and honest. This is the truth and you can believe me or not. Everything else said by a sick liar is wrong and only tries to frame others.

Well. . . that is quite a mouthful of information. According to this new person, GnosticPlayers is not one person, but actually two people.

And apparently Nclay was the person who hacked all of the sites, resulting in them adding DDB to the group. Yet if this person is *not* Nclay, who is he? I thought the group only had two members? Now I suddenly count three.

If we are to believe this post, then Nclay was the sole person who hacked GateHub for 10 million dollars, and decided out of some strict moral code to share most of that with the other two partners.

Something's not adding up for me. . . .

NOTE It has surfaced that DDB is the person behind this GnosticPlayers2 post, which make sense since the author is trying to distance himself from the hacks. Contrary to what he is saying, there are many sites that DDB hacked himself, such as Quora. He also did not "join the group later", he and Nclay were always together.

A Mysterious Third Member

During this chapter's introduction, I mentioned that a mysterious member of the group's inner circle, who I had known for quite some time, suddenly made

himself known as a security researcher. The following is an excerpt from one of our conversations:

```
whackyideas25:      If you know his co-conspirators, they are not very
                    fond of eachother.
Argon:              DDB?
whackyideas25:      They will give you more information than you can
                    expect
Argon:              you think that's ddb?
whackyideas25:      DDB and the third.
Argon:              wait who is the third?
whackyideas25:      You really not close on the case buddy
whackyideas25:      Gnosticplayers is the main man
Argon:              i was not even looking into this until yesterday so
                    give me a second to catch up
whackyideas25:      He started to work with ddb back in around 2016
whackyideas25:      He admitted that even on his Raidforums post
Argon:              ok right but who is the 3rd partner? Omni?
whackyideas25:      When he first hacked his first companies such as
                    Dailymotion, 8Tracks, Zomato
whackyideas25:      Ddb got in touch with him to learn
whackyideas25:      Then few years later, ddb adopted NSFW which learned
                    eveyrthing from ddb
whackyideas25:      Now NSFW and Gnostic have had some feud
Argon:              yeah all that stuff i know
whackyideas25:      Because from Gnostic point of view, he is nothing
                    but a leech using his own method
whackyideas25:      to hack companies such as Flipboard, Doordash,
                    Poshmark
whackyideas25:      To summarize - that basically is the trio
whackyideas25:      Gnosticplayers, Ddb, NSFW
```

NSFW/Photon

At this point I knew Photon aka Russian (who most people refer to as NSFW because they are a team) was working with the group, but I wasn't sure if he was an active member.

Before we go making wild accusations, why don't we see what NSFW has to say about this? The following is a conversation I had with him (under the name "Russian") on July 30, 2019:

```
Argon:              you mentioned you sold DDB some new s**t
Argon:              anything good that i dont have?
Russian:            well
Russian:            all of it was private stuff ye
Russian:            and if he finds out i sold to u that'd be pretty tragic
Russian:            since we're now partnered on hacking s**t
```

Fantastic. We have direct confirmation from NSFW that he is partnered with DDB, and by extension Nclay/GnosticPlayers. We also know from the previous chapters in this book that NSFW/Russian is responsible for hacking Flipboard, Poshmark, DoorDash, MGM, CodeChef, Timehop, and many, many other sites.

The Gloves Come Off

In a very unexpected turn, user OnSecurity (a random new user) responds to another user's message on Nclay's post with "SHUT THE F**K UP VINNY TROIA."

From there, yet another new player, K3l0t3x, jumps in (Figure 20.4).

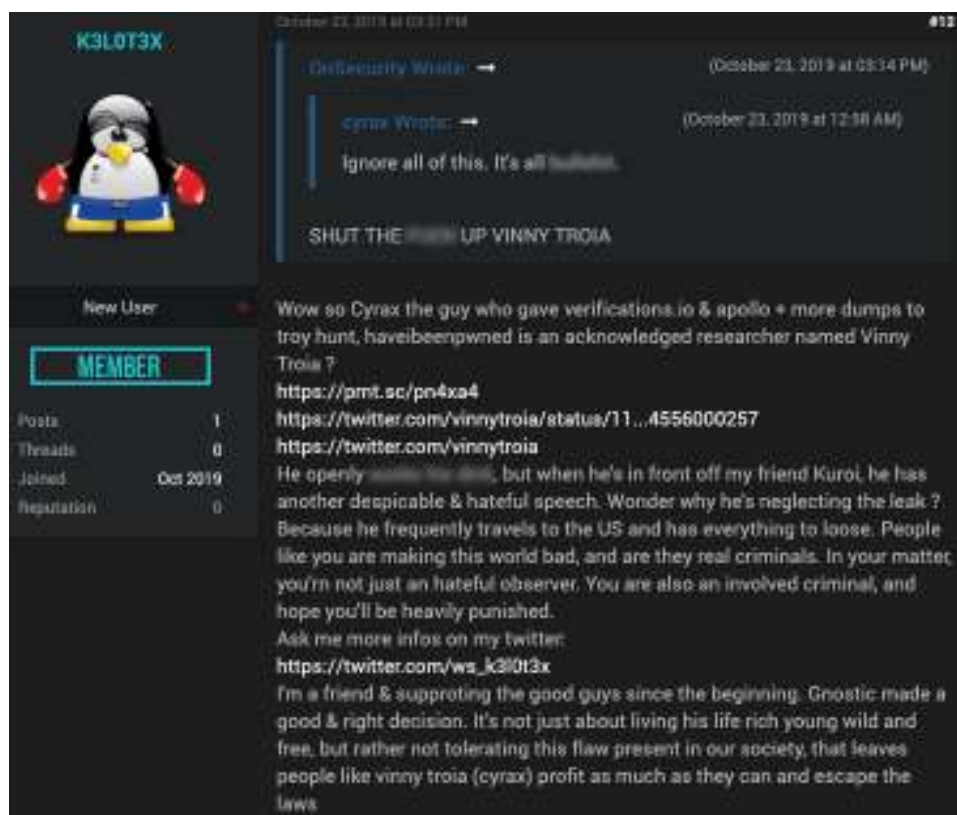


Figure 20.4

This entire post is nonsense, starting with the idea that I am Cyrax (because I am not). Regardless, I can tell Nclay wrote it because he has said similar things to me in the past.

The intent of this post was obviously to catch my attention and get me looking into this new user name. The post even goes so far as to say to contact him via Twitter, so let's take him up on that.

Making Contact

Now that I am involved in this little saga, I decided to do a little research. This works out for everyone, especially you, the reader, because what better way to bring together all of the different investigative techniques that we have learned throughout this book than to look into a group of cyber criminals claiming to have stolen 10 million dollars in XMR?

The first thing I did was go back and look through all of my old conversation logs, and I found something interesting—on February 12, 2019, NSFW sent me the following dox for Nclay (the same one that is now being leaked):

```
Username : Kuroi'sh, nclay, shg_amar, amar_shg, irbl00d
First Name : Gabriel
Last Name : Kimiaie-Asadi Bildstein
Age : 19
Country : France
City : Tarbes
Address : 6 rue emile raysse
Zip : 65000

Email :
flam6@protonmail.com
carradio@protonmail.com
miraiever@protonmail.com
gaby.tarbes@gmail.com

Mother :
Phone Number : 06 08 47 98 50
Work: Ophthalmologist
Work address: 24 rue larrey
City: Tarbes
Postal Code: 65000
Work number: 05 62 93 29 29 | 05 62 34 99 93

Siege: MADAME LAURE BILDSTEIN
Since: 04-05-1998
STREET: 41829912900019
Email : laure.bildstein@orange.fr
```

It would appear that they were planning this for a long time, and NSFW, for whatever reason, is the one who has been feeding me the disinformation. Let's examine the information I am receiving.

Gabriel/Bildstein aka Kuroi'sh

"Gabriel K. A. B.," as reported by Variety, is an 18-year-old French citizen. He and his partner, Prosox, were charged with several criminal counts of fraudulent data modification regarding the hacking of the "Despacito" music video on YouTube.

A quick search for Kuroi'sh reveals plenty of information on his YouTube hack, and also leads to his Twitter (@kuroi_dotsh) and GitHub (github.com/securitygab) pages.

Now that Gabriel/Kuroi'sh is suddenly not having any problems revealing to the world that he is Nclay, he should have no problem admitting it to me.

So I decided to reach out.

NOTE At this point, the rumor is still being spread of Nclay's death. If this person really committed suicide, he should not answer.

Twitter user Kuroi'sh (@kuroi_dotsh) responded right away. Unfortunately, I was not able to get anywhere with this person and was not even really sure if Gabriel was even in control of this account. Figure 20.5 shows a part of our conversation.

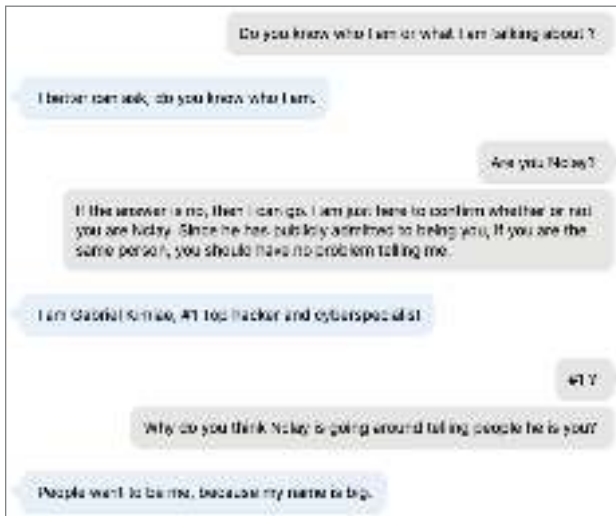


Figure 20.5

If this were really Nclay, he would not have responded this way. He knows exactly who I am and his English is much better.

Gabriel's official Twitter account turned out to be a dead end (for now), so we can move on.

Next, Kuroi'sh's GitHub page references his Zone-H public defacement page (<http://www.zone-h.org/archive/notifier=Kuroi>). As mentioned earlier in this book, Zone-H is a historical archive of public website defacements.

Website defacement hackers always include shout-outs to their friends. It is essentially an archive of skid behavior, which is why this is a perfect place to start looking.

When we look through the Zone-H archives, we can see hundreds of website defacements by Kuroi'sh, all with calling cards to the following people:

Greetz : Kolotex, RxR, Prosox, General KBKB, Shade, Sxtz

Contacting His Friends

The next thing I decided to do was look up his friends on Twitter. The defacements go back far enough where I should be able to find someone who knows the actual Kuroi'sh. I reached out to the following people:

- Who Am I/k3l0t3x (@ws_k3l0t3x)
- RevSec (@cyb0rg_fs)
- 9-4 Boy (@FuegoLevel)
- Prosox (@ProsoxW3b)
- Chic000 (@Chic000w3b)

NOTE To be fair, I did not know Chic000 even existed until he took it upon himself to send me a DM over Twitter with the message “mind your business.”

RevSec (@cyb0rg_fs) was the first to respond and seemed to know Gabriel very well. Figures 20.6 and 20.7 show the highlights.



Figure 20.6

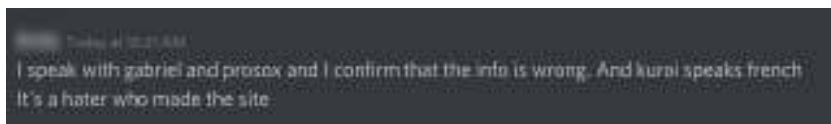


Figure 20.7

So now we have someone who actually knows Gabriel telling us that all of the information we have been reading is fake.

The plot thickens. . . .

Weeding through Disinformation

Regarding the “site” referred to in Figure 20.7, I asked this person about a very interesting WordPress blog that I came across. I will say that whoever thought up this plan certainly put a lot of time and effort into developing disinformation.

I suppose when you are talking about getting away with robbing millions of dollars in cryptocurrency, you want to make sure your bases are covered.

The WordPress blog, titled “Nassim AKA Prosox From Morocco,” can be viewed here:

<https://prosoxrealnamenassimfrommorocco.wordpress.com>

The blog was made by someone wanting to expose Prosox as being the “dumbest hacker ever.” To ensure their search engine keywords are covered, they even created a page called “Kuroi’S&H & Friends.”

The page is full of conversations that are allegedly from Kuroi’s&H, and it appears that the last blog post was written on March 14, 2017. Everything looks completely legitimate and aged well enough to cast reasonable suspicion.

Again, I call bulls**t.

Verifying with Wayback

A WordPress blog that has been up since March 2017 should have some sort of archive on Wayback Machine. This site does not.

There are, however, two URLs captured by Wayback (Figure 20.8).

2 URLs have been captured for this domain.

URL	NAME	TIME	FROM	TO	CAPTURED	DUPLICATES	UNRESOLVED
https://prosoxrealnamenassimfrommorocco.wordpress.com/	prosoxrealnamenassimfrommorocco.wordpress.com	Feb 19, 2018	Mar 24, 2018	2	15	3	
https://prosoxrealnamenassimfrommorocco.wordpress.com/kuroi'sh & friends/	kuroi'sh & friends	Aug 1, 2018	Aug 1, 2018	1	1	1	

Showing 1 to 2 of 2 entries

Find Previous 1 Next Last

Figure 20.8

The first file is `ar.txt`, which was captured in April 2018. The second is a URL with a Kuroish tag that looks like it was published August 2019. The `robots.txt` file is empty, and there is no `sitemap.xml` file, which makes this even less believable.

The site *currently* has a complete `sitemap.xml` file (as do most WordPress sites), so if these blog posts were really published in 2017, Wayback would have cached a copy of the site. Instead, we can see that the tags were added two years later.

We can file this one under “completely false.”

With this detour behind us, let’s pick up where we left off and continue down the path of contacting Kuroi’sh’s former associates.

Bringing It All Together

Going back to RaidForums, the post that specifically calls me out (Figure 20.4) was written by user K3l0t3x (let’s call him Kel). I can say with certainty that Nclay wrote it. The writing style is the same as his, and he even said a few of those *exact* things to me in a chat a few days earlier.

So now the question remains: Why would he use Kel’s name? The poster knew that I would want to speak with this person, which is why he told me to contact him via Twitter.

So I did. And whoever answered seemed to know a lot about what was going on (Figure 20.9).

According to this person, Nclay (Kuroi’sh) was scammed out of the 4 million dollars by some Russian. This makes more sense as to why he would create such a public scene.

And just to make sure we were all talking about the same person, he confirmed Kuroi’sh’s jabber account as “outofreach” (Figure 20.10).

Outofreach@jabber.ua is the jabber address that I have always used to communicate with Nclay. There is no question about that, which means whoever this person is knows what is going on and seems to be trying to further the story of Gabriel being Nclay.

Since people close to the real Kuroi’sh are telling me that all of this is fake, it would not be unreasonable to say that anyone trying to further the lie is a part of it.

Data Viper

Since Kel is now an obvious person of interest, let’s see what we can find out about him. The first thing I did was head over to Data Viper and look up his username (Figure 20.11).



Figure 20.9



Figure 20.10

🔍	IndoChina	R3LCT1X	malloyedto@gmail.com
🔍	cyberdb	R3LCT1X	anonymous.30m@gmail.com
🔍	darkcooper	R3LCT1X	anonymous.30m@gmail.com

Figure 20.11

We have two email address hits. Starting out with the first email address, `anonymous.30.m@gmail.com`, gives us several new pieces of information to explore (Figure 20.12).

Username	Email	Password	IP Address
R3LCT1X	anonymous.30.m@gmail.com		
	anonymous.30.m@gmail.com		
	anonymous.30.m@gmail.com		📍 88.169.240.198
R3LCT1X	anonymous.30.m@gmail.com		
h3y0x7	anonymous.30.m@gmail.com	80577h3y0x7	📍 37.150.151.200
R3LCT1X	anonymous.30.m@gmail.com		
R3LCT1X	anonymous.30.m@gmail.com		

Figure 20.12

IP addresses are always a good hit to explore, especially when they are not a VPN or proxy (which this is not). Expanding our search to `88.169.240.198` gives us a nice range of new matches, all seemingly related to the same person (Figure 20.13).

Username	Email	Password	IP Address
h3y0x7	44mehostree@outlook.fr	96577h3y0x7	📍 88.169.240.198
M0t1y5	44mehostree@gmail.com	abbyblue14	📍 88.169.240.198
	anonymous.30.m@gmail.com		📍 88.169.240.198
DeF1x	44mehostree@outlook.fr		📍 88.169.240.198
M0t1y5	44mehostree@gmail.com	abbyblue14	📍 88.169.240.198
Fl00d	h3y0x7@gmail.com		📍 88.169.240.198

Figure 20.13

Each of these email addresses and usernames are valid, so it is important to keep track of what you are doing. We can also see that there is a common password of `abbyblue14`, which reveals even more usernames and common IPs (Figure 20.14).

Username	Email	Password	IP Address
	savoieimattys@gmail.com	as10yb1ucl4	
Heavenly	hpmad@gmail.com	as10yb1ucl4	
Mattys	savoieimattys@gmail.com	as10yb1ucl4	# 88.165.240.198
	savoieimattys@yahoo.fr	as10yb1ucl4	
	hpmad@gmail.com	as10yb1ucl4	
	savoieimattys@yahoo.fr	as10yb1ucl4	
	savoieimattys@yahoo.fr	as10yb1ucl4	
Heavenly	hpmad@gmail.com	as10yb1ucl4	
Mattys	savoieimattys@yahoo.fr	as10yb1ucl4	# 88.165.240.198
	savoieimattys@yahoo.fr	as10yb1ucl4	

Figure 20.14

NOTE A mindmap tool (or Maltego) is a great way to visualize these connections so you can backtrack if you need to.

As we follow this long trail of accounts, usernames, and passwords, it appears that all roads eventually lead to two email addresses: mattysdbz@gmail.com and savoieimattys@gmail.com.

Based on this information, it would appear that K0l0t3x's real name is Matty Savoie.

That seems a bit too easy, though.

Trust but Verify

I am extremely skeptical, and a threat actor able to carry out this level of deception would have a long string of accounts that lead to incorrect information.

Now that we have some information to tie back to our person, let's see what kind of information we can gather using password reset clues. Starting with the email address anonymous.30.m@gmail.com, a PayPal password reset shows us a partial phone number (Figure 20.15).

Next, checking the password reset information in Google for the email savoieimattys@gmail.com gives us a different prompt (Figure 20.16).

What I love about this particular screen is that we can see that the email address is tied to an active phone number. We can also see that the number ends in 34, which is most likely the same number used in the PayPal phone verification for anonymous.30.m@gmail.com.

But what I really love about this verification method is that we can actually use it to verify whether we have the correct phone number.

If we enter an incorrect number here, Google will tell us and will not send the verification code. So once we have a number that we think is accurate, all we need to do is come back here and put in the number to see if it's valid!



Figure 20.15

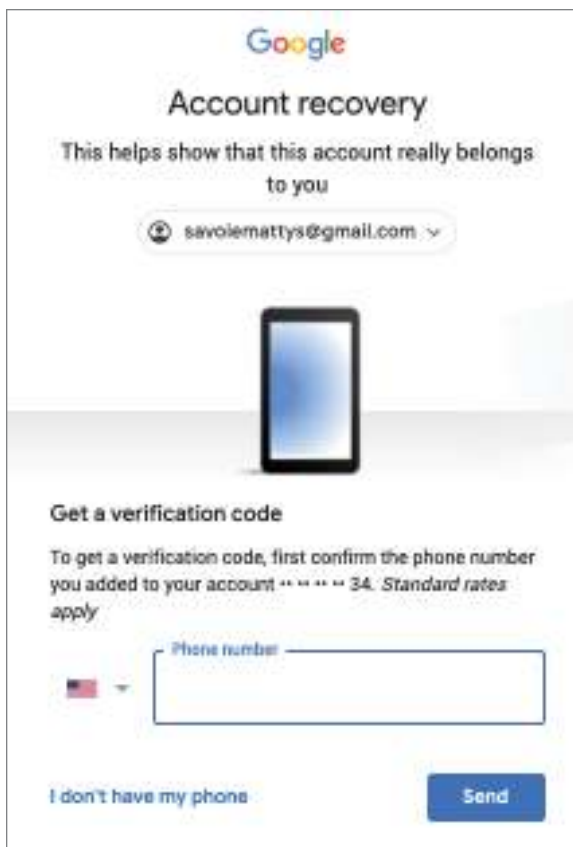


Figure 20.16

PayPal was kind enough to give us a lot of information to work with, but 07
• • • • 1 34 is not enough. Let's keep looking.

Domain Tools' Iris

The next step is to see if we can find any domains registered to the emails we have. There is always a good chance you will get a hit with threat actors, which is why Domain Tools' Iris is such a critical piece to my investigative workflow.

Running a search for anonymous.30.m@gmail.com gives us an immediate hit!

Domain Name: nova-stresser.net
Registrant Name: K*** W****
Registrant Organisation:
Registrant Street: 30900 *****
Registrant City: Nimes
Registrant State/Province:
Registrant Postal Code: 3*****
Registrant Country: FR
Registrant Phone: +33.78xxxxx34
Registrant Phone Ext:
Registrant Fax: +43.xxxxxxxx
Registrant Fax Ext: 1
Registrant Email: anonymous.30.m@gmail.com
Name Server: ns1.easyname.eu
Name Server: ns2.easyname.eu

We now have a name, address, and phone number, which looks amazingly similar to the phone number we need to verify both the PayPal and Google accounts.

NOTE I am masking his personal information because I have no way of knowing if this person is actually involved or not, and I am not in the business of doxing people.

Running another search for the phone number gives us a second match (Figure 20.17).



Figure 20.17

On this screen, we have a new email address (keloattacker@gmail.com), and we also have two possible names, K*** W**** and Ta**** La****, both with different addresses.

A Google search for “Ta**** La****” takes us right to LocateFamily.com (Figure 20.18).

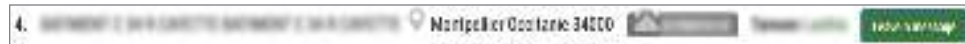


Figure 20.18

Verifying with a Second Data Source

Even with all of the love and effort that I put into ensuring Data Viper has as much data as possible, I know it is impossible to have *everything* covered. I have stressed many times throughout this book that you should never rely on a single source for your information, and my own tool is no exception to that rule.

WeLeakInfo.com is a website that allows anyone to look up a person’s password or other personal information. Similar to older sites like LeakedSource.com or Abusewith.us (which have all since been taken down by law enforcement), WLI allows anyone the ability to access this data for a small monthly fee.

Sites like this, while highly illegal, can still offer very important pieces of information. Use them while you can, because these sites don’t seem to last long.

In this case, I found an important match (Figure 20.19).

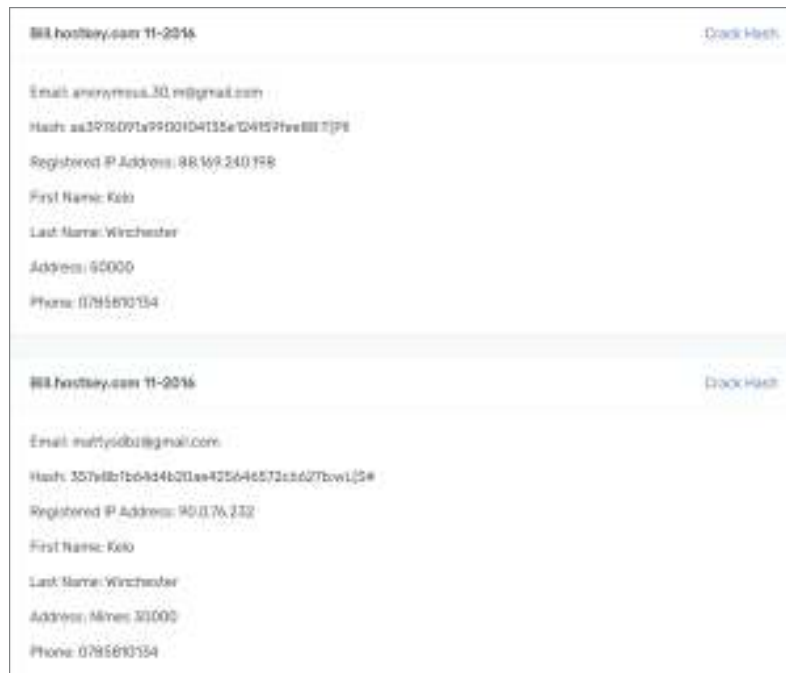


Figure 20.19

Data from a 2016 hostkey.com hack shows two different accounts with our matching phone number and email addresses. Since this is a hosting company, it would stand to reason that the name on this account would be valid.

Now we have two possible names for K3l0t3x. Of course it is possible that both names are fake. Either way, the phone number is still legitimate and active.

The End of the Line

Unfortunately, this is where my GnosticPlayers story comes to an end. This was never meant to be a complete attribution of the group members. Even though one of the people closely involved in GnosticPlayers was formerly a member of TDO, this was never my investigation. My attribution on NSFW was already finished when this saga happened.

At the very least, I have done my part and provided attribution on someone who, in my opinion, either is operating within the group structure or knows enough about the GateHub hack to provide sufficient information to lead to the right people. Whichever agency is investigating the GateHub hack can take this from here.

To be completely honest, I had no intention of even including Gnostic in my book until they dragged me into this mess.

It does make you wonder, though. . .

Why would Nclay intentionally drag me into this? He knew it would piss me off, so was *that* his plan all along? If Nclay wanted me to investigate Kel, it makes sense that he would call me out using his name.

But why?

What Really Happened?

The short answer is, I don't know. But now that we have the backstory on GnosticPlayers and the GateHub hack, here are some potential theories.

Outofreach

On February 11, 2019, TheRegister reported that a hacker known as GnosticPlayers posted three full rounds of exclusive databases for sale on Dream Market.

According to a subsequent article on ZDNet, "Gnosticplayers took credit for the hacks and denied being just an intermediary."

Sound familiar?

Around the same date, NSFW revealed the name of this mysterious seller to me as "Nclay," DDB's partner. This is the moment when I learned that DDB was just the seller, and Nclay was the hacker behind the hacks.

I reached out to the seller (as NSFW predicted I would), who introduced himself using the jabber address `outofreach@jabber.ua`.

And with that, the very first thing I learned about Nclay was that he was the hacker behind all the hacks, and that he also goes by Outofreach. Everything was beautifully spoon-fed to me.

Kuroi'sh Magically Appears

As if he were somehow telepathically summoned, Kuroi'sh contacted me on Twitter, using the account `@kuroi_dotsh`, which I previously dismissed. This time he seemed more eager to talk and to convince me of who he was.

By this point I was honestly ready to be done with this entire saga, but there was one thing he said that really caught my attention because I had already been thinking it (Figure 20.20).

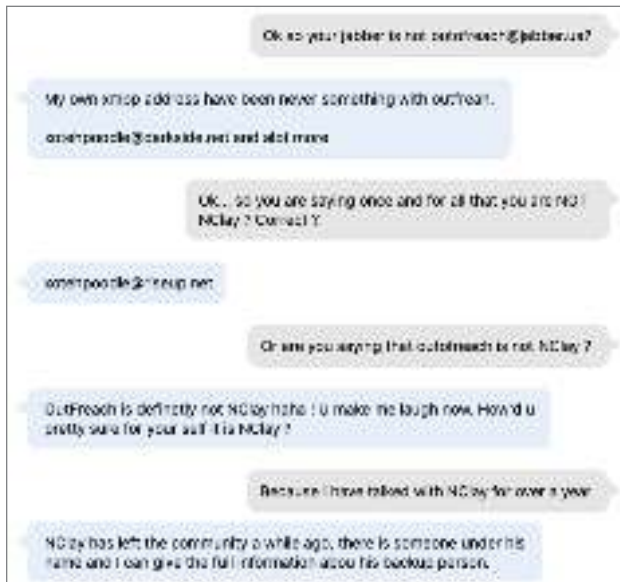


Figure 20.20

This person claiming to be Gabriel/Kuroi'sh seems to know a lot about Nclay, the most important detail confirming that Nclay and Outofreach (the person I *believed* to be Nclay) were not the same.

What I Learned from Watching Lost

Lost is one of my all-time favorite TV shows. One of the things I especially loved about the show was that you never knew who or what to believe. After watching it a second time, it occurred to me that the reason I was having such a

hard time figuring out what was happening was that I had a *natural inclination to assume everyone was lying*.

It's really interesting to rewatch the show with that perspective. You will quickly find that Benjamin Linus (one of the most mysterious and hated characters on the show) *never actually lied about anything*. He always told the truth about what was happening, but people chose to not believe him.

When watching the show for the first time, I assumed the "others" were lying about everything, which caused me to spend so much time trying to figure out what was really happening that I missed many obvious truths.

The same can be applied to this scenario.

At this point, there is so much information coming at me suggesting that Gabriel Nclay and Kuroi'sh are the same person that I can either continue to assume that everyone is lying or go with it.

Lightbulb moment:

I'm not sure why it never occurred to me that there could be more than one person using the name Nclay.

Who Hacked GitHub?

If we are to believe *everyone*, Gabriel Bildstein (aka Nclay aka Kuroi'sh) hacked the site (along with a few others).

The following is an excerpt of a private conversation between me and Russian (aka NSFW) on June 22, 2019:

```
Russian:      but this github hack added a lot of new players i hear
Russian:      i recently heard LE contact AWS, Github and Proxyrack
               regarding the matter
Russian:      and spoke to Github security, they have some leads on
               their group
Russian:      it is 4 people and they mixed it quite well
Russian:      we'll see if they run into issues
Russian:      depends on whether github is spending a lot or opened
               additional lawsuits
Russian:      i dont want to be in the blind
Russian:      i have been with them for a while i need to know whats up
Argon:        been with who? ddb/nclay?
Russian:      ddb
Russian:      nclay is a ni****
```

NSFW clearly knew who was responsible for the GitHub hack and said it was four "new people."

What an incredible coincidence that I just happened to be conversing with four new people over Twitter, all seemingly connected to Gabriel, Nclay, and this entire situation (K3l0t3x, RevSec, Chic000, and Kuroi'sh).

Unraveling the Lie

If we are to finally believe that Gabriel had anything to do with the GateHub hack, we should ask ourselves *why* we even know this.

The answer is because someone named GnosticPlayers publicly announced that he hacked GateHub. The next obvious question is, why would someone publicly admit to stealing all of that money?

The answer is: They wouldn't.

Other than the public confession, I have not seen any evidence to suggest that GnosticPlayers (aka Outofreach) was actually responsible for hacking GateHub.

This is the lie. *What better way to offload all of your crimes onto someone else?*

There is now so much information linking Gabriel/Nclay to GateHub that when Europol questions him they will undoubtedly assume that Nclay is the same Nclay that is a known part of GnosticPlayers. And why not, considering the fact that the Nclay's confession came from someone with the username GnosticPlayers?

Suddenly, all of the 50+ high-profile hacks associated with GnosticPlayers will be charged to someone else.

Very, very well done.

Was Gabriel Involved? My Theory

The truth is, I have no idea, but my gut tells me "no," because I learned of the name Nclay in February, which was about a month before the GateHub hack. In June, NSFW clearly stated that he knew who was behind the GateHub hack and attributes it to four "new people." In the same breath, he also admits to working with DDB, which tells me that they could have easily come up with this plan together (or were even working together all along).

Does it seem reasonable that six people would conspire to pin a 10-million-dollar robbery onto one person? Of course it does. That is a lot of money, and NSFW, by way of his history with TDO and Hell forum, already has a long history of pinning crimes on other people.

By his own admission, Gabriel used the name Nclay several years ago when he hacked Zomato and Edmodo, and the name has not been around since. This created the opportunity for someone to use the name and associate it to someone else with a known history of hacking.

Gabriel is Nclay: An Alternate Theory

The more I read conversations between the group members, the more I believe that Gabriel might actually be Nclay. Some of the things he is telling me can't

be denied, including information that only Gabriel should know — including which officer in the French Ministry of Justice is handling his case, and that four of his cars have been confiscated (including his 2 lamborghinis), as well as roughly 1.8 million dollars in Bitcoin has also been confiscated from him.

In addition, he has told me that when he first went to the police, he was sent to a hospital for having a mental breakdown. Perhaps this was his plan?

I have read numerous conversations between threat actors stating that Gabriel is “schizophrenic” or “Bipolar” (those are their words), and these medical diagnosis are how he was able to get away with previous hacks.

Perhaps he does actually have a medical condition which is how he knew he would walk away from a full confession.

Of course, this is all just speculation based on his own account of the situation . . . but the statements he has provided are accurate.

All roads lead back to NSFW

Regardless of Nclay’s motives or his true identity, the underlying connection to Photon (NSFW) can not be denied.

The following is a private conversation between two people close to this situation:

```
Digi:      jimmy and vinny are planning against nclay
Cyrax:     you are being silly
Digi:      okay
Cyrax:     im the one who talks to nclay
Digi:      ohhh
Cyrax:     nclay freaked out and thought i was vinny
Cyrax:     vinny has nothing to do with nclay
Digi:      really?
Cyrax:     100%
Digi:      who is vinny on rf?
Cyrax:     Bishop
Digi:      he is the one who posted about nclay on twitter
Cyrax:     OMG
Cyrax:     about the talk??
Cyrax:     thats what this is all been about??
Digi:      yes
Cyrax:     f**king idiots
Digi:      see
Digi:      you dont know anything
Cyrax:     there were no identities mentioned in his talk
Cyrax:     lol
Digi:      coz of vinny only it all started
Cyrax:     if that's the case
Cyrax:     then i know why this really started
Cyrax:     and his name has 4 letters in it
Digi:      am not quite sure about this
```

```
Digi:      omni?
Cyrax:     N S F W
Digi:      lmao
Digi:      yes
Digi:      nsfw and ddb
Cyrax:     f**king nsfw
Digi:      or you can say rawdata
Digi:      or photon
Cyrax:     rawdata is ddb
Digi:      i know that
Cyrax:     photon is nsfw
Digi:      thats why i said you can call them anything
```

Given the timing of everything that has happened, and the fact that NSFW has been the one spoon-feeding me much of this information, I can't help but question whether or not they are the same person. Regardless, this book and my TDO report will provide enough attribution on NSFW that I have no doubt the truth will come out once he is in custody.

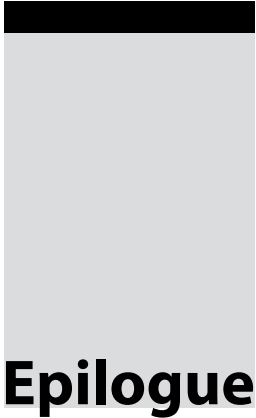
Until then, this saga has made for an excellent (and very educational) ending to this book!

Summary

Aside from the incredibly entertaining story, the main takeaway from this chapter is that nothing should be accepted as fact without verification. Cyber criminals will have no problem cooking up intricate backstories just to throw you off the trail, especially when there is money involved.

Just like anything else, if something seems too good to be true (e.g., like a criminal suddenly confessing to the theft of 10 million dollars), it usually is. Don't accept it as fact just because there appears to be legitimate conflict behind it.

Trust no one, verify everything, and don't assume that everyone is lying!



Epilogue

At the time of writing this Epilogue, Nathan Wyatt (who I believe is Arnie) has been extradited to the United States, and the other two primary members of The Dark Overlord are not far behind.

Sadly, the Canadian justice system moves *very* slowly. However, things *are* moving, and I will share with you how on know.

On November 22, 2019, I received the following email from someone that sounded very familiar.

You know who I am. I can only speak via this email: `vinnytroia@mailbox.org`, due to the position you put me in, you degenerate.

I am speaking to assess the damage you have done that I am not aware of.

I was very intrigued. Two days later, we setup a time to speak over Jabber under the alias “thegoodoldNSA”.

thegoodoldnsa: Well here's what I know.

thegoodoldnsa: Or, what I can infer.

thegoodoldnsa: You told the F.B.I and others about me, and what I've done.

thegoodoldnsa: You may have thought this was for not cooperating with you, but to ruin someone's life (potentially), is barbaric.

thegoodoldnsa: What did I do that warrants something serious, as to make sure I would be not only investigated by one, but multiple agencies?

thegoodoldnsa: And I can never get away from that conspiracy charge as I'm sure you're aware.

thegoodoldnsa: I am certain there's investigations, I am very
certain you have had a part in this.
thegoodoldnsa: For reasons I cannot discuss.
thegoodoldnsa: Those who told me are bound by law.

thegoodoldnsa: How someone I didn't deem a threat, turns out to be
the biggest.
thegoodoldnsa: Really something isn't it?
thegoodoldnsa: I just want you to realize that what you've done is
f**k my life.
thegoodoldnsa: And the only thing I want to understand now, is why
you did this to me?
thegoodoldnsa: We spoke for a long time, I acted like a d**k, but
it was warranted, you don't win by playing nice.
thegoodoldnsa: But seriously. What you've done is extreme.
--
Argon: It doesnt matter if I have the real TDO or not.
Wyatt is singing like a bird
Argon: he keeps talking a lot about the kid he trained
thegoodoldnsa: You've told me all I need to know. I wish I could
kill Wyatt now, f**k.

It would appear as though the FBI was questioning this person (who I believe to be TDO/NSA/NSFW based on his language style).

Regardless of who I thought it could be, this person was clearly upset, wanting to know why I would give any information to the FBI. . .

Then the next day I received a WhatsApp message from our friend Chris Meunier.

Chris Meunier: Stop playing games
Vinny Troia: what games am i playing?
Chris Meunier: You're pretending you don't know why I messaged you
Chris Meunier: I'm not here to cause you grief I just want to make
sure we get all the facts straight
Vinny Troia: well, funny enough, TDO messaged me 2 nights ago
Vinny Troia: or someone involved with those guys
Vinny Troia: telling me i am f**king with his life
Vinny Troia: and now here you are asking to help clarify facts
Chris Meunier: And that leads you to what conclusion?
Chris Meunier: That you're correct in what you're going to do?
Vinny Troia: or that something caused him to message me out of
the blue
Chris Meunier: You're doing something
Chris Meunier: Involving me
Vinny Troia: and i haven't talked to you in what? 6 months? and
now here you are
Chris Meunier: Look you want me to just give you a clue
Vinny Troia: that would be really helpful
Chris Meunier: It involves the FBI

Vinny Troia: yes we are on the same page. i spoke to the FBI
briefly about Nclay

Chris Meunier: Yeah I know you spoke to the FBI

Vinny Troia: why would you know that i spoke to the FBI?

Chris Meunier: We're both in contact with the FBI alright?

What an incredible coincidence that I receive a random message from someone who I happen to believe is TDO regarding the FBI speaking to him about me, then literally the next day Chris messages me saying the same thing.

If that weren't enough, the following is a snippet of a Twitter message sent to me from my friend Bev Rob. Apparently Chris also contacted her out of the blue to see if she had been speaking with the FBI.



What an interesting coincidence. What are the odds that someone sounding exactly like TDO would contact me and then Chris would contact me one day later asking the same question?

Where Do We Go From Here?

I was hoping to write this epilogue saying that this case was all wrapped up, but it seems nothing can rush the Canadian justice system. At the very least, it is obvious from those messages that there is some movement.

I suspect things will really start to pick up speed as we get closer to this book's publication date, and with any luck this story will be all wrapped up by the time you are reading this.

If things are not resolved by then, I suspect the people involved will be trying to shift as much of the blame as possible onto their former partners.

I have seen what happens on hacker forums when skids start arguing with each other. It never takes long for the truth to come out and to quickly escalate into a free-for-all with every man, woman, and child only looking out for themselves. The result is that everyone ends up going down in a giant blaze of glory.

Now that I think about it, that's probably what the law enforcement agencies have been planning all along.

It's how I would let it all play out.

In Closing, Thank You!

Either way, I want to thank *you* for purchasing my book. This book represents the sum of several years worth of investigative work, and I am grateful that you were interested in reading about my adventures in Hunting Cyber Criminals.

The challenges that I have experienced while investigating TDO were truly life changing in how they formed my thought processes for future endeavors, and I find it extremely humbling that you would want to learn about the different tools and techniques that I used to solve the many problems I encountered along the way.

With the release of this book also marks the launch of my Data Viper platform. It has become an amazing repository of actor-specific data and pre/post-breach threat intelligence. I am extremely excited about the possibility of seeing other organizations benefit from the years of hard work I have poured into developing this platform by helping them bring attribution to their crimes, and even to use the data to stop crimes before they occur.

Thank you again purchasing my book. If you find this book interesting, please let me know the next time we run into each other at a conference or feel free to send me an email or DM on Twitter.

Peace.

NUMBERS AND SYMBOLS

0-day, 276

- (minus sign) modifier, dorks,
160

A

AdvancedBackgroundChecks,
Skiptracer, 361

aliases, 20

allintitle: operator, dorks,
162

allinurl: operator, dorks, 165

Amazon, buckets, 293

Amazon S3 buckets, 320–321
CSF (CloudStorageFinder) and,
299–300

NoScrape and, 312

Ancestry.com, 331–332

API keys, 6

Apollo.io, 311

Archive.org, 214

search URLs, 217–219

archives. *See also* Wayback
Machine

CachedView.com, 212–214

search engine caches, 212–214

ARIN (American Registry of
Internet Numbers), 47–48

Arnie, TDO and, 32–35

CraftyCockney, 32

asset discovery, 46–47

ARIN (American Registry of
Internet Numbers), 47–48

Censys, 56

Subdomain finder, 56–57

DNSDumpster, 49–52

dorks, search engines,
48–49

Enumall, 59–60

Fierce, 57–58

Hacker Target, 52–53

Shodan, 53–55

Sublist3r, 58–59

tool results, 60–61

authentication, Elasticsearch
and, 311

AXIS webcam, Inurlbr, 172

B

- ban lists, forums, 381
- basic scrape, 313
- BeenVerified, 330, 331
- Bezlica.top, 216
- Bing, dorks, 169
- BitBucket, 280
 - SearchCode.com and, 281
- Bitcoin, 13
 - addresses, 15
- Bitcoin Cash, 13
- Bitcoin Talk forum, 274
- BlackBox forum, 275, 276–277
 - Cyper, 32
- blockchain, 13
- blockchain explorers, 13–15
- Blue Snap breach, 23
- breaches, 21–23
 - Blue Snap, 23
 - Regpack, 23
- browsers, user-agent field (HTTP header), 125
- bruteforce
 - credentials, 102
 - Wfuzz, 146
- bucket_finder.rb, CSF, 299–300
- BuiltWith, 121–122
 - Google Analytics ID, 123
 - IP history, 124
 - Relationship Profile tab, 123
 - Technology Profile tab, 122

C

- cache: operator, dorks, 165
- CachedView.com, 214
- caches, search engines, 212–214
- Canary Tokens, 25
- Carhart, Leslie, 25–26

- CAs (Certificate Authorities), 201
- CassandraDB, 301
 - NoScrape and, 312, 318–319
- Censys, Subdomain finder, 57
- Cewl, 10
- Chainalysis, 17
- Cimpanu, Catalin, 442
- Ciphertrace, 17
- Cloudflair, 209–211
- Cloudflare, 191, 204–209
 - Cloudflair, 209–211
- CMSMap, 129
 - batch mode, 130–131
 - scans
 - multiple sites, 130–131
 - single site, 130
 - vulnerability detection, 131–132
- CMSs (content management systems)
 - parameters, 129
 - WIG and, 124–125
- code repositories
 - BitBucket, 280
 - GitHub, 280
 - GitLab, 280
 - SearchCode.com, 281–282
 - false negatives, 283–284
- code reuse, 282
- Coggle.it, 198
- Coinfirm, 17
- command-line tools, MongoDB, 305–308
- communication style, TDO, 30
- Computer Misuse Act, Wyatt, Nathan, 33
- configuring, Linux, 5–6
- Coral Cache, 214

- Cr00k, 35–36, 403, 446. *See also*
 - NSFW; Peace/Peace of Mind
 - aliases, 35–36
 - Data Viper and, 422–429
 - forum search, 379
 - identifying, 422–431
 - Karvouniaris, Dennis, 451
 - search results, 367–368
 - TDO and, 407–408
 - Userrecon screen, 370–372
- Cran, Jonathan, 95, 96
- credentials, bruteforce, 102
- Cree.py, geolocation, 343–346
- Crenshaw, Adrian, 433
- criminal records, state court
 - web sites, 332–333
- crt.sh, 204
- CRUD (create, read, update, delete) rights, 295
- Crunch, 10
- cryptocurrencies, 4, 11–12
 - Bitcoin, 13
 - blockchain explorers, 13–15
 - Chainalysis, 17
 - Ciphertrace, 17
 - Coinfirm, 17
 - Elliptic, 17
 - investigations, 15–17
 - exchanges, 17–18
 - traders, 17–18
- CSF (CloudStorageFinder), 298–299
 - Amazon S3 buckets, 299–300
 - parameters, 299
 - bucket_finder.rb, 299–300
 - Digital Ocean and, 299, 300–301
 - downloading, 299
 - public URLs, 299
 - space_finder.rb, 299, 300–301
 - r all, 300
 - spider_finder.rb, 299
 - SpiderOak, 299
- CSV files
 - Cree.py export, 343–346
 - dorks, 162
 - geolocation information, 343–346
- CT (Certificate Transparency), 201
- Cloudflare, 204–209
 - Cloudflair, 209–211
- crt.sh, 204
- Google project, 202
- logs, 206–208
- CTFR, 202–203
 - subdomains, 203–204
- CURL commands, OSINT.rest, 351
- Curvve Recordings, 328
- cyber attribution, 25
- cybercriminals, 26–27
- Cyper (CyPeRtRoN), 31–32, 267, 276
 - aliases, 32
 - Argon story, 383–386
 - BlackBox forum and, 276–279
 - Dark Overload group, 30
 - Hackweiser, 290
 - identity trace, 278–280
 - KickAss Forum and, 84, 388
 - KickAss Framework, 287
 - social engineering and, 383–387
 - TinEye, 336

- Twitter page, 337
- WHOIS, 182
- Wikipedia, 291
- Cypertron, 32

D

- darkweb marketplaces, 388
- data, attribution, 295–296
- data dumps, 413
 - collections 1–5, 413–414
 - locating data, 420
 - quality, 413–414
 - locating, 420
 - verifying, 415–419
- Data Viper, 407
- Cr00k, 422–423
 - discovery, 427–429
 - forums, 423
 - timeline analysis, 423–427
- data curation, 413
- DEVON and, 430–431
- forums, 421–422
- Kel and, 470–472
- PHP, 420
- ReactJS, 420
- databases
 - downloading results, 296
 - NoSQL, 301
 - ransoming, 311
- DataViper, 275
- DerbyCon, 433
- DEVON, Data Viper and, 430–431
- Diachenko, Bob, 293–294
- diff tools, Wikipedia, 291
- DigiNotar, 201–202
- Digital Ocean, 293
 - CSF (CloudStorageFinder) and, 299, 300–301
- Dirhunt, 143
 - FADAA, 144–145
 - parameters, 144
- distributed scanning
 - proxy chains, 73
 - proxy service, rotating, 73
- DNS map, Photon-generated, 152
- DNSDumpster, 49–52
 - Photon and, 150
- DNSTwist, 61–62
- DOC files, dorks, 162
- document metadata, 245
 - Exiftool, 246–247
 - FOCA (Fingerprinting Organizations with Collected Archives), 261–262
 - metadata extraction, 263–266
 - project start, 262–263
- Intrigue.io, 257
 - bucket information, 259–260
 - URI Spider, 257–259
- Metagoofil, 248–250
- Recon-NG modules
 - Interesting_files, 252–254
 - Metacrawler, 250–252
 - Pushpin, 254–257
- documents, MongoDB, 302
- Dogecoin, 13
- domains
 - CT (Certificate Transparency), 202
 - reverse IP lookup, 189
 - searches, 188
 - WHOIS searches, 177
 - advanced, 181–182
 - bulk parsed, 189
 - keywords, multiple, 179–181

- namedroppers.com, 177–183
 - threat actors, 182–183
 - DomainTools, 177, 188
 - domain searches, 188
 - Domain Status field, 195
 - IP addresses, reverse lookup, 189
 - Iris, 221–222, 474–476
 - Hosting History tab, 232–234
 - Pivot Engine, 222–229
 - screenshot history, 232
 - search bar, 222
 - WHOIS History tab, 230–231
 - screenshots, 193
 - WHOIS
 - bulk parsed, 189
 - cross-checking information, 197–199
 - history, 192, 193–195
 - Reverse WHOIS, 196–197
 - View Whois Record, 190–191
 - dorks, search engines, 48–49, 159
 - (minus sign) modifier, 160
 - Bing, 169
 - DualxCrypt.org, 166
 - Exploit-DB, 159
 - Inurlbr, 169–173
 - operators
 - allintitle:, 162
 - allinurl:, 165
 - cache:, 165
 - filename:, 165
 - filetype:, 162–163
 - intext:, 165–166
 - intitle:, 161–162
 - inurl:, 163–165
 - site:, 161
 - quotes, 160
 - Yahoo!, 169
 - DualxCrypt.com, 84
 - VirusTotal, 87–88
 - dump parameter, Elasticsearch, 317
- ## E
- EagleEye, 333, 340–343
 - Elasticsearch, 293, 301
 - curl command, 308–309
 - databases, downloading, 296
 - JSON, 308–310
 - NoScrape and, 312
 - dump, 317
 - keyword search, 315
 - matchdump, 317–318
 - scan, 314–315
 - search, 315–317
 - passwords, 409–413
 - querying, 308
 - Shodan and, 296
 - Elliptic, 17
 - email, Skiptracer searches, 361–364
 - emails
 - address discovery, 156
 - Photon, 150
 - enum_wayback (Ruby script), 215–216
 - Ethereum, 13
 - Exactis, 294, 311
 - exchanges, cryptocurrencies, 17–18
 - Exiftool, 246–247
 - exit scam, 388
 - Exploit-DB, dorks, 159
- ## F
- Facebook
 - OSINT.rest, 352–357
 - password reset, 393–394

FADAA (Florida Alcohol and Drug Abuse Association), 144
false negatives, SearchCode.com, 283–284
Fierce, 57–58
filename: operator, dorks, 165
files, Photon, 150
filetype: operator, dorks, 162–163
FIN scans, 68
 results, 71
firewalls, 70
 address spoofing
 IPs, 78–79
 MAC, 78–79
 checksums, 76–77
 data length changes, 78
 decoy scan, 77
 distributed scanning
 proxy chains, 73–74
 proxy service, rotating, 73
 TOR, 73–74
 firewalking, 79
 fragmented packets, 74
 MTU (maximum transmission unit), 74
 service detection, 74–75
 timing of requests, 76
FOCA (Fingerprinting Organizations with Collected Archives), 261–262
 metadata extraction, 263–266
 project start, 262–263
forums, 274
 ban lists, 381
 BitcoinTalk, 274
 BlackBox, Cyper, 32
 Data Viper and, 421–422

HackForums.net, 379
 history, 275–278
 KickAss, Cyper, 32
 Nulled.to, 379–380
 OGUsers.com, 379
 RaidForums.com, 379
fragmented packets, 74
FreeBackgroundCheck.org, 330, 331
full scrape, 313
Fuller, Rob, 214–215
fuzzing, 147

G

Gabriel/Buildstein (Kuroi'sh), 465–467, 477
Galaxy 2, 289, 291
GateHub hack, 453, 478–480
GDPR (General Data Protection Regulation), 176
geolocation, Cree.py, 343–346
Ghost, 32
Git commit logs, 287–288
GitHub, 280
 SearchCode.com and, 281
Gitlab, Night Lion Security, 312
GitLab, 280
 SearchCode.com and, 281
Gitrob, 284–287
Gmail, password reset, 391–393
Gnostic Players, 409
GnosticPlayers, 454–456, 463–476
 hacking techniques, 457–459
 NSFW, 463
 posts, 459–461
 sites hacked, 457
GnosticPlayers2, 461–462

Google
 CT project, 202
 reverse image search, 333, 357
 search engine caches, 212–214
 Google Analytics, BuiltWith and, 123
 Google Code, SearchCode.com and, 281
 Google Dorks
 operators
 allintitle:, 162
 allinurl:, 165
 cache:, 165
 filename:, 165
 filetype:, 162–163
 intext:, 165–166
 intitle:, 161–162
 inurl:, 163–165
 site:, 161
 queries, 159
 Google Hacking Database. *See* Exploit-DB
 Google Images, 334–336
 Greenberg, Andy, 295
 Guided Pivots (Iris), 223–224
 Historical Search, 224–225
 IP, 225
 IP address, 226
 Name Server, 225
 Registrant Organization, 225
 SSL certificate hashes, 227–229

H

Hacker Target, 52–53
 HackForums.net, 379
 Hackweiser, 289
 Cyper, 290

 members, 291
 Wikipedia site, 290
 Hadnagy, Chris, 324, 382
 Hardigree, Steve, 295
 HavelBeenPwned, Skiptracer, 361
 Heid, Alex, 160
 Hell Forum, 35, 275
 Hell Reloaded, 32
 Historic Whois Lookups, 186
 HL7 medical software, TDO and, 29–30
 homoglyphs, 61
 Honey Badger, 25
 Hunchly, 9
 Hunt, Troy, 414, 419

I

ICANN (Internet Corporation for Assigned Names and Numbers), 176
 ICQ, password reset and, 403–404
 IDS (intrusion detection system), 70
 image searches, 333
 EagleEye, 333, 340–343
 Google Images, 334–336
 reverse image lookup, 357
 Google reverse image search, 333
 reused profile pictures, 339
 TinEye, 333, 336–340
 ImageRaider, 340
 images, saving from social media, 357
 Instagram, password reset, 400

- IntelTechniques, 7–8
 - tasks, 95–96
- interactions with threat actors
 - drawing out, 433–434
 - information flow
 - establishment, 446
 - hacker drama and, 447–452
- Obfuscation/Stradinatras, 436–439
- TDO, 450–451
- WhitePacket, 434–440
- YoungBugsThug, 440–446
- intext: operator, dorks, 165–166
- intitle: operator, dorks, 161–162
- Intrigue.io, 84, 95, 257
 - bucket information, 259–260
 - EmailAddresses entities, 156
 - entities, 95
 - Entities tab, 96–99, 154–155
 - machines, 95
 - results
 - analysis, 104–105
 - export, 105–107
 - Run Task, 152–154
 - spider module, 152–153
 - uberpeople.net analysis, 99–103
 - URI Spider, 152–153, 257–259
- inurl: operator, dorks, 163–165
- Inurlbr, 169–173
- investigations, 19
 - Carhart, Leslie, 25–26
 - cybercriminals, 26–27
 - paths, 25–26
- IP addresses
 - Guided Pivots (Iris), 226
 - historical information queries, 297
 - history, BuiltWith, 124

- owner identification, 295
- reverse lookup, DomainTools, 189
- tracking, 346–347
- WPScan, 138
- IP attribution, 46
- Iris, 221–222, 474–476
 - Hosting History tab, 232–234
 - Pivot Engine, 222
 - Guided Pivots, 223–224, 225–229
 - Historical Search, 224–225
 - IP address, 226
 - Screenshot History tab, 232, 235, 240
 - search bar, 222
 - WHOIS History tab, 230–231

J

- Java API, Elasticsearch, 308
- jQuery, password reset and, 400–403
- JSON
 - Elasticsearch, 308–310
 - WIG output, 126–128

K

- Karvouniaris, Dennis, 451
- Kayak, 331
- Kaye, Daniel, 442–443
- Kel, Data Viper and, 469–472
- keywords, multiple, WHOIS, 179–181
- KickAss forum, 275, 284
 - closing, 388
- Cyper, 32
 - social engineering and, 383–387

L

letter substitutions, username searches, 369
 LinkedIn, Skiptracer, 361
 Linux, configuration, 5–6
 Litecoin, 13
 logs, Git commit logs, 287–288
 Lucidchart, 198

M

MABNA, 409
 Maltego, 5
 SocialLinks, 350–351, 358–360
 Martin, William, 73–74, 311
 matchdump parameter,
 Elasticsearch, 317–318
 metadata. *See also* document metadata
 Metagoofil, 248–250
 Metasploit
 enum_wayback (Ruby script), 215–216
 Recon-NG and, 108
 Wayback Machine scraping, 214–215
 Meunier, Christopher, 435
 OSINT.rest, 355–357
 Microsoft, password reset, 399–400
 mind mapping tools, 198
 MindBendingBeats.ca, 237
 Mindnode, 198
 Mirai malware, 442
 MITM (man-in-the-middle) attacks, 202
 MO (modus operandi), 27
 modules, Recon-NG
 hacker target, 114

list, 108–109
 searches, 111
 Shodan, 115–116
 using, 111–115
 virustotal, 113, 114–115
 MongoDB, 293, 301
 command-line tools, 305–308
 commands
 find(), 307
 find().pretty(), 307
 show dbs, 307
 use, 307
 database connection, 305–306
 documents, 302
 NoScrape and, 312, 313
 basic scrape, 313
 full scrape, 313
 ransomed server, 313–314
 Robot 3T tool, 302–305
 Connect button, 303
 PrivateDB, 303–304
 RobotMongo, 302
 morality, 24
 MTU (maximum transmission unit), 74
 Murdock, Cat, 327, 333–334
 MySpace, Skiptracer, 361

N

namedroppers.com, 177–179
 Nclay, 453–454
 GnosticPlayers site hacks, 458
 Raid Forums, 456
 Ndiff, 79
 NeoBoss, 426
 Night Lion Security, Gitlab page, 312
 NightCat, 435–436

NightSquare, 288

NMAP

--mtu, 74

--proxy, 73

--reason, 71

-A, 74

-badsum, 76

-f, 74

-O, 74

-p-, 68

-scanflags, 68, 69

-sn, 68

-sS, 69

-sU, 68, 69, 74

-sV, 74

-sW, 70

-sX (Xmas scan), 68, 69

-T, 76

-T3, 76

-T4, 76

FINPSH, 69

firewalls, 70

checksums, 76–77

data length changes, 78

decoy scan, 77

distributed scanning, 73–74

TOR, 73–74

firewalking, 79

fragmented packets, 74

IP address spoofing, 78–79

MAC address spoofing, 78–79

MTU (maximum

transmission unit), 74

service detection, 74–75

timing of requests, 76

Ndiff, 79

proxychains, 73

PSH, 69

reason response, live server
identification, 71–72

reports, 80

results comparisons, 79–81

scans, 67

active host list, 68

configuring, 68–69

FIN, 68

full port scans, 68–70

SYN, 68

TCP ports, 69

TCP windows, 70

types, 69

SYNFIN, 69

uberpeople.net, 100–101

NoScrape, 304, 311–312

Amazon S3 buckets and, 312,
320–321

CassandraDB and, 312, 318–319

Elasticsearch and, 312

dump, 317

keyword search, 315

matchdump, 317–318

scan, 314–315

search, 315–317

MongoDB and, 312

basic scrape, 313

full scrape, 313

ransomed server, 313–314

NoSQL databases, 293, 301

CassandraDB, 301

Elasticsearch, 301

data dump, 311

querying, 308–311

MongoDB, 301

command-line tools, 305–308

Robot 3T tool, 302–305

RobotMongo, 302

NSA (Peace of Mind), 32, 36, 374,
445–446
aliases, 38
hostility, 37–38
Louisiana Dept of Motor
Vehicles, 36
NSFW, 463, 476–477, 480
Nulled.to, 379–380

O

Obfuscation/Stradinatras,
436
TDO and, 437–439
OGUsers.com, 379
Open Source Intelligence tools,
6
open-source tools, 5
operators, dorks
allintitle:, 162
allinurl:, 165
cache:, 165
filename:, 165
filetype:, 162–163
intext:, 165–166
intitle:, 161–162
inurl:, 163–165
site:, 161
OPSEC, vanity and, 429
Orange is the New Black, TDO
(The Dark Overlord) and, 27
OSINT, 2
framework, 6
OSINT.link, 6
OSINT.rest, 350–351
CURL commands, 351
Facebook searches, 352–357
Run in Postman button,
351

Search Information by Phone,
351–352
Twitter, 357–358

P

packets, fragmented, 74
parameter-based URLs, Photon,
150
password managers, 409
passwords, 408–409
profile matrix, 409–413
recycling, 409
resetting, 382, 390–391
Facebook, 393–394
Gmail, 391–393
ICQ, 403–404
Instagram, 400
jQuery responses, 400–403
Microsoft, 399–400
PayPal, 394–397
Twitter, 397–399
verification sheet, 391
paste sites, 273
psbdmp.ws, 273–274
PayPal, password rest, 394–397
Peace of Mind. *See* NSA (Peace
of Mind)
people searches, PIPL, 326–330
PeopleLocker, 330, 331
Petya/NoPetya, 14
phishing domains, 61–64
DNSTwist, 61–62
phone numbers, Skiptracer
searches, 364–366
Photon, 149, 237
DNS map, 152
information extracted, 150
parameters, 150

- performing a crawl, 151–152
- Wayback Machine scraping, 216–217
- PHP
 - Data Viper and, 420
 - Inurlbr, 172
- PIPL, 326
- Pivot Engine (Iris), 222
 - Guided Pivots, 223–224, 225–229
 - Historical Search, 224–225
 - IP, 225
 - Name Server, 225
 - Registrant Organization, 225
 - SSL certificate hashes, 227–229
 - Historical Search, 224–225
 - IP address, 226
- pivots, 222
- PrivateDB (MongoDB), 303–304
- profile matrix
 - F3ttywap, 409–413
- forums
 - ban list, 381
 - searching, 379–381
- proxies, Storm Proxies, 10–11, 138
- psbdmp.ws, 273–274
- public records
 - Ancestry.com, 331–332
 - FreeBackgroundCheck.org, 330
 - SkipEase.com, 330
- Python
 - Skiptracer
 - email address searches, 361–364
 - phone number searches, 364–366
 - username searches, 366–369
 - Userrecon, 370–372

Q

- queries, MongoDB, 302
- quotes, dorks, search engines, 160

R

- Raid Forums, 456
- RaidForums.com, 379
- ransomware, Petya/NoPetya, 14
- RDP (remote desktop protocol), 29
- ReactJS, 420
- Recon-NG, 84, 107–108
 - commands
 - add domains, 110
 - reverse resolve, 112
 - set source, 112
 - show domains, 110–111
 - show options, 112
 - databases, local, 110
 - home screen, 108
 - Metasploit and, 108
 - modules
 - hacker target, 114
 - hosts-hosts/resolve, 111–112
 - Interesting_files, 252–254
 - list, 108–109
 - Metacrawler, 250–252
 - Pushpin, 254–257
 - searches, 111
 - using, 111–115
 - virustotal, 113, 114–115
 - Shodan, modules, 115–116
- Reddit Investigator, 372–374
- regex patterns, Photon, 150
- Regpack, 23
- reports, NMAP, 80
- reverse IP lookup, 189
- Reverse WHOIS, 196

- Revolt, 374
 - Twitter search, 398
- Robb, Bev, 435
- Roberts, Chris, 274–275, 276
- Robot 3T (MongoDB)
 - PrivateDB, 303–304
 - screens, 302–303
 - Studio 3T, 302
- Robot 3T tool, Connect button, 303
- RobotMongo, 302
- Ruby, `enum_wayback` script, 215–216
- S**
- SANS, 325
- `scan` parameter, Elasticsearch, 314–315
- scans, NMAP, 67
 - active host list, 68
 - configuring, 68–69
 - FIN, 68
 - full port scans, 68–70
 - SYN, 68
 - TCP ports, 69
 - TCP windows, 70
 - types, 69
- scraping
 - basic scrape, 313
 - code repositories, 280–281
 - Git commit logs, 287–288
 - Gitrob, 284–287
 - SearchCode.com, 281–284
 - forums, 274
 - full scrape, 313
 - NoScape, 311–312
 - Amazon S3 buckets and, 312, 320–321
 - CassandraDB and, 312, 318–319
 - Elasticsearch and, 312, 314–318
 - MongoDB and, 312
 - paste sites, 273–274
 - theHarvester, 269–273
 - tools, URLs, 217–219
 - Wayback Machine, 214–215, 237–238
 - Photon, 216–217
 - site digest, 219–220
 - wiki sites, 288–289
 - Wikipedia, 289–292
- screenshots
 - DomainTools, 193–194
 - history, 232
- script-kiddies (skids), 274
- SCTs (signed certificate timestamps), 202
- search engines
 - caches, 212–214
 - dorks (*See* dorks, search engines)
 - search limit numbers, 270
- search parameter, Elasticsearch, 315–317
- SearchCode.com, 281
 - BitBucket, 281
 - false negatives, 283–284
 - Git commit logs, 287–288
 - GitHub, 281
 - GitLab, 281
 - Gitrob, 284–287
 - Google Code, 281
- SecLists, 9
- secret keys, Photon, 150
- service detection, 74–75
- Shodan, 53–55

- command-line options, 296
 - commands, parse, 296
 - Elasticsearch and, 296
 - modules, 115–116
 - queries, historical data, 296–298
 - results, downloading, 296
 - site: operator, dorks, 161
 - skids (script-kiddies), 274
 - SkipEase.com, 330, 331
 - Skiptracer
 - AdvancedBackgroundChecks, 361
 - email address searches, 361–364
 - HaveIBeenPwned, 361
 - LinkedIn, 361
 - MySpace, 361
 - phone number searches, 364–366
 - True People, 361
 - Truth Finder, 361
 - username searches, 366–369
 - SOC (security operations center), 25
 - social engineering, 381–382
 - Argon story, 383–387
 - KickAss forum and, 383–387
 - social media, 349
 - images, saving, 357
 - OSINT.rest, 350–351
 - Facebook searches, 352–357
 - Instagram, 354
 - Photon, 150
 - Reddit Investigator, 372–374
 - Skiptracer
 - email address searches, 361–364
 - phone number searches, 364–366
 - username searches, 366–369
 - TOR-based, 289
 - Userrecon, 370–372
 - SocialLinks, 350–351
 - Maltego users, 358–360
 - space_finder.rb CSF, 299, 300–301
 - spider_finder.rb CSF, 299
 - SpiderFoot, 84
 - access, 85
 - discovered elements list, 90–91
 - IP address scan, results, 86–87
 - uberpeople.net, 89–90
 - VirusTotal, 87–88
 - SpiderFoot HX, 91
 - results section, 92–93
 - SpiderOak, CSF
 - (CloudStorageFinder) and, 299
 - SSL certificates, Pivot Engine, 227–229
 - state court web sites, 332–333
 - Storm Proxies, 10–11, 138
 - Stradinatras, 436
 - Strand, John, 24–25, 325–326, 347, 386–387
 - subdomains
 - CT (Certificate Transparency) and, 202
 - CTFR, 203–204
 - Photon, 150
 - Sublist3r, 58–59
 - SYN scans, 68
- ## T
- TCP, window scan, 70
 - TDO (The Dark Overlord), 27–28
 - Bitcoin withdrawal, 451

- communication style, 30
 - forums and, 275–278
 - group structure, 30–31
 - Arnie, 32–35
 - Cr00k, 35–36
 - Cyper (CyPeRtRoN), 31–32
 - NSA, 36–41
 - healthcare hack, 32
 - KickAss forum closing, 388–389
 - leadership change, 36–37
 - Obfuscation/Stradinatras, 437–439
 - PIPL search, 329–330
 - presentation, 38–41
 - school terrorism, 37
 - victims, 28–29
 - TeamSkeet.com, 424–425
 - Termbin, 8
 - Thalet-Fischer, Maxime, 455
 - theHarvester, 268
 - arguments, 268–269
 - parameters
 - b, 270
 - d, 270
 - l, 270
 - scans, running, 269–273
 - threat actors. *See also* specific actors
 - interacting with, drawing out, 433–434
 - WHOIS searches, 182–183
 - Tim Tomes, 25
 - TinEye, 333, 336–340
 - Cyper, 336
 - tokens, Canary Tokens, 25
 - tools, open source, 5
 - TOR, 73–74
 - traders, cryptocurrencies, 17–18
 - True People, Skiptracer, 361
 - Truth Finder, Skiptracer, 361
 - Twitter
 - OSINT.rest, 357–358
 - password reset, 397–399
 - typosquatting, 61–64
 - DNSTwist, 61–62
- ## U
- uberpeople.net
 - Intrigue.io, 99–103
 - SpiderFoot, 89–90
 - URLs
 - Photon, 150
 - public, CSF and, 299
 - scraper tools, 217–219
 - user-agent field (HTTP header), 125
 - username
 - letter substitutions, 369
 - Skiptracer searches, 366–369
 - Userrecon, 370–372
- ## V
- verification sheet, password reset and, 391
 - Verifications.io, 311
 - View Whois Record (DomainTools), 190–191
 - VirusTotal, 87–88
 - vulnerabilities
 - CMSMap scans, 131–132
 - WordPress, plugins, 139–141
- ## W
- Wayback Machine, 211–212, 468
 - Photon, 150
 - scraping, 214–215, 237–238

- enum_wayback (Ruby script),
 - 215–216
- Photon, 216–217
- site digest, 219–220
- screenshots and, 193
- web crawlers
 - performing a crawl, 151–152
- Photon, 149
- Wfuzz, 146
 - 301/302 response codes, 148
 - output, 148
 - parameters, 146–147
 - typo3temp folder, 148
 - scanning, theHarvester, 271–272
- WhitePacket, 434–435
 - Bill, 439–440
 - Reddit Investigator, 373
 - Robb, Bev, 435
 - Twitter, 358
 - YoungBugsThug, 440–446
- WhitePacket.com, 184–185, 234–235
- WhitePages, 330, 331
- WHOIS, 47, 89, 175–176
 - cross-checking information, 197–199
 - data uses, 176–177
 - Domain Status field, 195
 - domains, searches, 177
 - advanced, 181–182
 - keywords, multiple, 179–181
 - namedroppers.com, 177–179
 - threat actors, 182–183
 - DomainTools, 177
 - View Whois Record, 190–191
 - Historic Whois Lookups, 186
 - historical, 177
 - history, 192, 193
 - ownership changes, 194–195
 - Reverse WHOIS, 196–197
 - WhitePacket.com, 184–185
 - Whoisology.com, 177, 183–184
 - advanced searches, 187
 - WHOIS History tab (Iris), 230–231
 - Whoisology.com, 183–184
 - advanced searches, 187
 - Historic Whois Lookups, 186
 - WhitePacket.com, 184–185
 - WhoisRWS, 47
 - WIG (Webapp Information Gatherer), 124
 - CMSs and, 124–125
 - file output, 126–128
 - JSON output, 126–128
 - parameters, 125
 - scanning all targets, 126
 - user agent masking, 125
 - wiki sites, 288–289
 - Wikipedia, 289–292
 - Wikipedia, 289–292
 - Compare Selected Revisions button, 291
 - diff tools, 291
 - View History page, 290
 - Wired, 295
 - Wood, Robin, 298
 - wordlists
 - Cewl, 10
 - Crunch, 10
 - SecLists, 9
 - WordPress

- plugins, vulnerabilities,
 - 139–141
- WPSan and, 136–141
- WPSan, 132
 - enumerate u, 138
 - force, 137
 - proxy, 137
 - stealthy, 137
- IP addresses, rotating, 138
- output, 134–136
- parameters, 133–134
- WAFs, 136–141
- WordPress not detected,
 - 136–141
- Wyatt, Nathan, 32, 349. *See also*
 - Arnie, TDO and
 - Computer Misuse Act, 33
 - Dissent Doe interview, 33

- extradition, 33–34
- indictment, 33–34
- OSINT.rest, 352–354
- YouTube audio, 34

X

- Xdedic, 29
- XLS files, dorks, 162, 163

Y

- Yahoo!, dorks, 169
- Yandex, 340
- YoungBugsThug, 433–434,
 - 440–446

Z

- ze0ring, 276
- ZIB trojan, 435
- Zynga, 459